

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Paulo Ricardo Torres Diniz

**Uma Aplicação do Processamento de Linguagem Natural
na Modelagem Conceitual de Sistemas de Informação**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Fernando Álvaro Ostuni Gauthier

Florianópolis, Abril de 2002

Uma Aplicação do Processamento de Linguagem Natural na Modelagem Conceitual de Sistemas de Informação

Paulo Ricardo Torres Diniz

Esta Dissertação foi julgada adequada para obtenção do título de Mestre em Ciência da Computação - Área de Concentração de Sistemas de Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Fernando Álvaro Ostuni Gauthier, Dr.

Fernando Álvaro Ostuni Gauthier, Dr.

Banca Examinadora

Fernando Álvaro Ostuni Gauthier, Dr.

João Bosco da Mota Alves, Dr.

Luiz Fernando Jacintho Maia, Dr.

Dedicatória

Dedico este trabalho a meu pai, Ilídio Silva Diniz, falecido em 1980, que apesar dos poucos anos que acompanhou meu desenvolvimento pessoal mostrou para mim a importância da perseverança e da honestidade na vida de um homem.

Agradecimentos

Primeiramente devo agradecer a Deus pela força que Ele sempre me dá, principalmente nos momentos onde o objetivo parece distante ou até impossível.

Não posso esquecer também de agradecer a minha esposa Marta, e meus filhos Tiago e Taiana pela paciência e compreensão que tiveram durante todo o desenvolvimento de minha pesquisa para conclusão desta dissertação.

Agradeço a minha mãe Vera pelo seu esforço na minha criação e incentivo para que eu jamais deixasse de estudar para poder alcançar grandes objetivos.

Agradeço também a UFSC - Universidade Federal de Santa Catarina pela oportunidade que me foi ofertada de cursar este programa de pós-graduação e ao Prof. Fernando Gauthier que foi compreensivo e paciente, demonstrando características dignas de um orientador, que cobra no momento oportuno e entende as necessidades de seus orientados.

Por último, mas não menos importante, agradeço aos alunos do curso de Tecnologia em Processamento de Dados da UNOPAR - Universidade Norte do Paraná por terem colaborado nos testes do software e, principalmente a aluna Renata Mota pelo auxílio dado na parte de programação do sistema e correção do sistema.

Resumo

O início do ensino da disciplina de Banco de Dados mostra-se muitas vezes frustrante, tanto para o aluno quanto para o professor, em virtude da falta de técnicas que auxiliem o discente a formatar textos de levantamento de dados, que se mostram, na maioria das vezes, dúbios e redundantes. Este trabalho visa rever as metodologias e técnicas existentes hoje para este objetivo, culminando em propor a criação de um software de auxílio que possibilite ao aluno desenvolver seu texto de levantamento de dados seguindo especificamente a técnica de Lista-Diagrama e ainda indique possíveis falhas, utilizando para isso a teoria do processamento de linguagem natural.

Abstract

The beginning of the education of discipline of Database reveals many times frustrating, as much for the pupil how much for the teacher, in virtue of the lack of techniques that assist the learning to format data-collection texts, that if they show, in the majority of the times, redundant.. This work aims at to today review the existing methodologies and techniques for this objective, culminating in considering the creation of software of assists that it makes possible the data-collection pupil to develop its text specifically following the technique of List and Diagram and still indicates possible imperfections, using for this the theory of the processing of natural language.

Sumário

ÍNDICE DE FIGURAS	10
<u>1. – INTRODUÇÃO</u>	<u>13</u>
1.1 - PROBLEMATIZAÇÃO	14
1.2 - OBJETIVO	15
1.3 - JUSTIFICATIVA	16
<u>2. - REVISÃO BIBLIOGRÁFICA</u>	<u>19</u>
2.1 - SISTEMAS DE INFORMAÇÕES.....	19
2.2 - PARADIGMAS DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO.....	21
2.2.1 - CICLO DE VIDA TRADICIONAL OU EM CASCATA.....	21
2.2.2 - ENGENHARIA REVERSA.....	22
2.2.3 - ENGENHARIA DA INFORMAÇÃO	23
2.2.4 - BUSINESS DESIGN.....	26
2.2.5 – ORIENTAÇÃO A OBJETOS	29
2.3 - SISTEMA OBJETO DE ESTUDO	31
2.4 - INTEGRIDADE DE DADOS E ANOMALIAS	34
2.4.1 – ANOMALIAS DE INCLUSÃO DE DADOS.....	35
2.4.2 – ANOMALIAS DE ALTERAÇÃO DE DADOS.	36
2.4.3 – ANOMALIAS DE EXCLUSÃO DE DADOS.	37
2.5 - DEPENDÊNCIAS FUNCIONAIS.....	38
2.5.1 - DEPENDÊNCIAS FUNCIONAIS SIMPLES. (DS).	40
2.5.2 - DEPENDÊNCIAS MULTIVALORADAS (DMV).....	42
2.6 - TÉCNICA LISTA DIAGRAMA	43
2.6.1 - PASSOS PARA PROJETO ATRAVÉS DE LISTA DIAGRAMA	43
2.6.2 - ATIVIDADES ASSOCIADAS AO 1º PASSO.	45
2.6.3 - T1.-REGRAS PARA CONSTRUIR LISTAS DE DEPENDÊNCIAS.	47
2.6.4 - ATIVIDADES ASSOCIADAS AO 2º PASSO	51
2.6.5 - T2.-REGRAS PARA OBTER UM DD A PARTIR DA LD.....	51
2.6.6 - ATIVIDADES DO 3º PASSO.....	57
2.6.7 - T3.-REGRAS PARA OBTER AS TABELAS BÁSICAS.....	57
2.6.8 - REGRAS PARA POSICIONAMENTO RELATIVO DAS TABELAS BÁSICAS DA BD.....	60
2.7 – INTELIGÊNCIA ARTIFICIAL	63
2.7.1 - PROCESSAMENTO DA LINGUAGEM NATURAL.....	65
<u>3. - PROPOSTA DE SOLUÇÃO</u>	<u>69</u>
3.1 - DEFINIÇÃO DAS REGRAS	69
3.2 - DEFINIÇÃO DOS PROCESSOS	69
3.3 - ENTRADA DE DADOS	70
3.4 - CADASTRAMENTO DAS REGRAS.....	75
<u>4. – PROTÓTIPO DO SISTEMA</u>	<u>84</u>

4.1 – DEFINIÇÃO DA BASE DE DADOS DE SUPORTE.	84
4.1.1 – TABELAS	84
4.1.2 – RELACIONAMENTOS	85
5. - PROJETO	86
5.1 SOLUÇÕES PROPOSTAS PARA O SISTEMA	86
5.2 DOCUMENTAÇÃO DA ANÁLISE	86
5.2.1 LISTA DE USE CASES	87
5.2.2 LISTA DE EVENTOS	88
5.2.3 - DIAGRAMAS DE USE CASE	89
5.2.4 USE CASES	91
5.2.4 DIAGRAMA DE CLASSE	144
5.3 DOCUMENTAÇÃO DE PROJETO	144
5.3.1 DESCRIÇÃO DAS DECISÕES DE PROJETO	145
5.3.2 REFINAMENTO DOS DIAGRAMAS NA FASE DE ANÁLISE	145
5.3.3 MAPEAMENTO DO DIAGRAMA DE CLASSE PARA O BANCO DE DADOS RELACIONAL	150
5.3.4 DESCRIÇÃO DA PADRONIZAÇÃO DO SISTEMA	151
5.3.5 PROTOTIPAÇÃO DA INTERFACE	152
5.4 DOCUMENTAÇÃO DA IMPLEMENTAÇÃO	161
5.4.1 DESCRIÇÃO DA LINGUAGEM DE PROGRAMAÇÃO UTILIZADA	161
5.4.2 DESCRIÇÃO DA FERRAMENTA UTILIZADA PARA CRIAÇÃO DO HELP ON-LINE	161
6. – TRABALHOS FUTUROS APÓS CONCLUSÃO	162
7. – CONCLUSÃO	163
BIBLIOGRAFIA	164
ANEXOS	161
ANEXO A – IMPRESSÃO DO LAY-OUT DAS TELAS E RELATÓRIOS	162
CADASTRAR SISTEMA	162
CADASTRAR REGRAS	162
CADASTRAR PALAVRAS	163
CADASTRAR PALAVRAS FIXAS	163
CADASTRAR PALAVRAS NÃO ATÔMICAS	164
CADASTRAR SINÔNIMOS	164
VISUALIZAÇÃO DO RELATÓRIO DE REGRAS	165
IMPRESSÃO DO RELATÓRIO DE REGRAS	166
IMPRESSÃO DO RELATÓRIO DE REGRAS ORIGINAL	166
RELATÓRIO DAS PALAVRAS NÃO ATÔMICAS	167
RELATÓRIO DAS PALAVRAS	168
RELATÓRIO DOS SINÔNIMOS	168
RELATÓRIO DOS SISTEMAS	169
ANEXO B – MANUAL DE INSTALAÇÃO	170

<u>UTILIZANDO BANCO DE DADOS ACCESS</u>	<u>171</u>
<u>UTILIZANDO BANCO DE DADOS ORACLE</u>	<u>173</u>
<u>ANEXO C – MANUAL DO USUÁRIO</u>	<u>175</u>

ÍNDICE DE FIGURAS

Figura 1 – Modelo Clássico de Abstração segundo MACHADO & ABREU (1995).....	14
Figura 2 - Interação entre as fases da EI segundo FELICIANO NETO & FURLAN & HIGA(1988)	25
Figura 3 - Ciclo de Vida Business Design.....	27
Figura 4 - Enfoque baseado em Sistema X Enfoque baseado em Objeto, segundo FURLAN (1998)	30
Figura 5 - Modelo de Abstração a partir do Sistema Objeto de Estudos segundo FELICIANO NETO & FURLAN & HIGA(1988).....	32
Figura 6 - Representação das Visões do SOE	33
Figura 7 - Representação de modelo Lógico segundo FELICIANO NETO & FURLAN & HIGA(1988).....	34
Figura 8 - Base de dados para exemplo de anomalia de inclusão.....	36
Figura 9 - Base de dados para exemplo de anomalia de alteração.....	37
Figura 10 - Base de dados para exemplo de anomalia de exclusão	37
Figura 11 – Representação de Dependência Multivalorada	42
Figura 12 - Modelo de Abstração em Lista Diagrama segundo Baños (1997).....	44
Figura 13 – Representação de Ciclo (Deadlock)	54
Figura 14 – Representação de alvos com recepção múltipla de arcos.....	56
Figura 15 – Representação de Bandeira de Domínio	56
Figura 16 – Representação de Diagrama de Dependências.	58
Figura 17 – Representação de Diagrama de Dependências com chave terminal.....	58
Figura 18 – Representação da marcação de leitura no diagrama de dependências.....	60
Figura 19 - Exemplo de Diagrama de Dependências	61
Figura 20 - Modelo de Tabelas Básicas Interligadas.....	62
Figura 21 - Protótipo da Tela de Cadastramento de Sistema.....	71
Figura 22 - Protótipo da Tela de Cadastramento das Regras do Negócio	73
Figura 23 - Protótipo da Tela de Reordenação das Regras do Sistema	74
Figura 24 - Protótipo do Relatório de Regras cadastradas no sistema.....	75
Figura 25- Diagrama de Dependências do Problema Exemplo	79
Figura 26 - Modelo Relacional da Base de Dados de Suporte	85
Figura 28 – Use Case de Apoio ao Sistema.....	89
Figura 30 - Use Case 01 – cadastrarSistema – Curso Normal.....	91
Figura 31 – Diagrama de Colaboração – cadastrarSistema – Curso Normal.....	92
Figura 32 - Use Case 01 – cadastrarSistema – Curso Alternativo 1	93
Figura 33 – Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.....	94
Figura 34 - Use Case 01 – cadastrarSistema – Curso Alternativo 3	94
Figura 35 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 3	95
Figura 36 - Use Case 01 – cadastrarSistema – Curso Alternativo 4.....	96
Figura 37 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 4	96
Figura 38 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.1	97
Figura 39 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.1	97
Figura 40 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3.....	98
Figura 41 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3	99
Figura 42 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3.....	99
Figura 43 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3	100
Figura 44 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3.1	100
Figura 45 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3.1	101
Figura 46 - Use Case 02 – cadastrarPalavra – Curso Normal.....	102
Figura 47 – Diagrama de Colaboração – cadastrarPalavra – Curso Normal	103
Figura 48 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1.....	104
Figura 49 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1	105
Figura 50 - Use Case 02 – cadastrarPalavra – Curso Alternativo 3.....	105
Figura 51 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 3	106
Figura 52 - Use Case 02 – cadastrarPalavra – Curso Alternativo 4.....	107
Figura 53 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 4	107

Figura 54 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1.3.....	108
Figura 55 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1.3	109
Figura 56 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1.3.....	109
Figura 57 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1.3	110
Figura 58 - Use Case 03 – cadastrarRegra – Curso Normal	111
Figura 59 – Diagrama de Colaboração – cadastrarRegra – Curso Normal.....	112
Figura 60 - Use Case 03 – cadastrarRegra – Curso Alternativo 1	113
Figura 61 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 1	114
Figura 62 - Use Case 03 – cadastrarRegra – Curso Alternativo 3	115
Figura 63 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 3	115
Figura 64 - Use Case 03 – cadastrarRegra – Curso Alternativo 3.2	116
Figura 65 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 3.2.....	116
Figura 66 - Use Case 03 – cadastrarRegra – Curso Alternativo 1.2	117
Figura 67 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 1.2.....	117
Figura 68 - Use Case 04 – cadastrarSinonimo – Curso Normal	118
Figura 69 – Diagrama de Colaboração – cadastrarSinonimo – Curso Normal.....	119
Figura 70 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1	120
Figura 71 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1	121
Figura 72 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 3	121
Figura 73 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 3	122
Figura 74 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 4.....	123
Figura 75 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 4	123
Figura 76 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1.3	124
Figura 77 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1.3	124
Figura 78 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1.3	125
Figura 79 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1.3	126
Figura 80 - Use Case 05 – cadastrarPalavraFixa – Curso Normal	127
Figura 81 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Normal	128
Figura 82 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1.....	129
Figura 83 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1	130
Figura 84 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 3.....	130
Figura 85 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 3	131
Figura 86 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 4.....	132
Figura 87 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 4	132
Figura 88 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1.3.....	133
Figura 89 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1.3	134
Figura 90 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1.3.....	134
Figura 91 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1.3	135
Figura 92 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Normal	136
Figura 93 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Normal	137
Figura 94 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 1	138
Figura 95 – Diagrama de Colaboração– cadastrarPalavraNaoAtomica – Curso Alternativo 1	139
Figura 96 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 3	139
Figura 97 - Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 3.....	140
Figura 98 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 4	140
Figura 99 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 4	141
Figura 100 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3	142
Figura 101 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3	142
Figura 102 - Use Case 06 – cadastrarPalavraNaoAtomica – Caso Alternativo 1.3.....	143
Figura 103 - Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3.....	143
Figura 104 - Diagrama de Classes de Análise	144
Figura 105 - Diagrama de Use Case – Cadastrar Regras.....	148
Figura 106 - Diagrama de Use Case – Cadastrar Sistema	148
Figura 107 - Diagrama de Use Case – Cadastros da Base de Apoio	149
Figura 108 - Diagrama de Use Case - Relatórios	149
Figura 109 – Mapeamento do Diagrama de Classes.....	150
Figura 110 - Menu Principal do Sistema	152
Figura 111 – Tela de cadastro do sistema.....	153

Figura 112 – Tela de Cadastramento de Regras	154
Figura 113 – Tela de Atualização de Regras do Sistema.....	155
Figura 114 – Tela de reordenação das regras do sistema	155
Figura 115 – Tela de cadastro de palavras de dicionário do sistema.....	156
Figura 116 – Tela de cadastro de palavras não atômicas.....	157
Figura 117 – Tela de cadastro de sinônimos	157
Figura 118 – Tela de palavras das áreas fixas do cadastro de regras do sistema.....	158
Figura 119 – Mensagem de Criação de um novo sistema.	158
Figura 120 – Mensagem de Erros para sistema com nome duplicado.....	159
Figura 121 – Mensagem de efetivação de alteração do sistema.	159
Figura 122 – Tela para entrada de parâmetro de busca de sistema.....	159
Figura 123 – Tela de erro para parâmetro de busca não encontrado na base de dados.....	159
Figura 124 – Tela de erro para campo autor não preenchido no cadastro do sistema.	160
Figura 125 – Tela de informação sobre exclusão do sistema.	160
Figura 126 – Tela de informação de erro de integridade referencial entre sistema e regras.....	160

1. – Introdução

Pode-se dizer que a modelagem de dados, inicialmente proposta por Peter P. Chen na década de 70 é a técnica de representação que serve para mostrar o conjunto de conceitos do ambiente através dos conjuntos de dados a ele relevantes, além dos relacionamentos entre esses conjuntos.

Ao definir o modelo de abstração atualmente utilizado para o projeto de sistemas de banco de dados, a maioria dos autores da área de banco de dados faz referência a uma modelagem de dados dividida em três camadas: Conceitual, Lógica e Física. Essa técnica propõe-se a fazer uma pesquisa mais formal sobre os dados, sua organização, sua origem e seus significados, aplicados a um determinado contexto. Para uma melhor didática, os níveis de abstração podem ser conceituados da seguinte forma:

Nível Conceitual: é o nível de abstração mais elevado onde os dados são descritos através de estruturas simples (entidades) e os relacionamentos entre essas estruturas. No ensino de banco de dados, este nível comumente se confunde com o nível lógico e apresenta um alto grau de dificuldade inicial para que o aluno o compreenda de imediato, visto que sempre que um levantamento de necessidades é feito, este é descrito em linguagem natural escrita, a qual não tem uma formalização definida. Este modelo não tem como objetivo obter-se informações sobre a organização física dos dados, mas sim as necessidades conceituais do sistema e a organização natural dos dados na realidade observada.

Modelo Lógico: o modelo lógico é derivado do modelo conceitual: É utilizado para especificar a estrutura lógica geral do banco de dados e para fornecer uma diagramação de alto nível do que será implementado. Neste nível também há vários modelos e o enfoque comum é dado ao modelo relacional. Este modelo é baseado na representação de dados e relacionamentos entre dados por um conjunto de tabelas, cada uma possuindo um número de colunas com nomes únicos.

Modelo Físico: é o nível de abstração mais baixo. Descreve como os dados então realmente armazenados. Em um nível físico, complexas estruturas de dados de baixo nível são descritas em detalhes.

A figura abaixo representa os passos que são seguidos na modelagem do mundo real, ou seja, da realidade dos negócios, usando a implementação de um banco de dados.

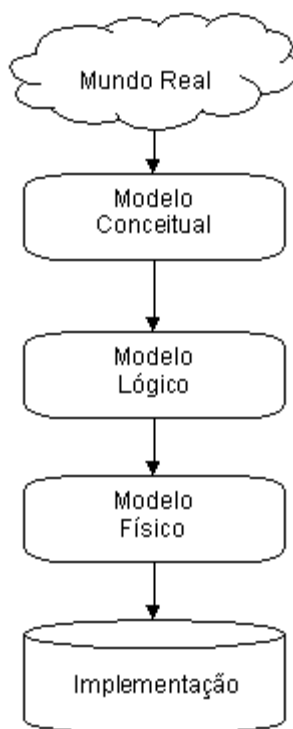


Figura 1 – Modelo Clássico de Abstração segundo MACHADO & ABREU (1995)

1.1 - Problematização

A criação de uma Base de Dados consistente, confiável e eficiente permite o desenvolvimento de sistemas de informação íntegros e seguros, porém muitos analistas desprezam esta regra e constroem simples armazenadores de dados, desprezando por vezes os fortes recursos existentes hoje nos Sistemas Gerenciadores de Banco de Dados do mercado. Isso se dá, principalmente pela falta de entendimento do sistema na etapa de levantamento de dados e análise, visto que poucos autores discutem esta etapa em técnicas de base de dados, mas sim em níveis de projeto de sistemas, que é uma etapa muito importante para o desenvolvimento de um sistema, mas que não pode desprezar a base de

dados, pois nela estarão as estruturas necessárias para o desenvolvimento e que, na falha delas deverão ser substituídas por pesados algoritmos incorporados ao código do sistema, semelhantes aos utilizados na época de desenvolvimento de sistemas baseados em arquivos.

No ensino da disciplina de banco de dados nota-se uma grande falha na bibliografia existente, onde pouco se fala de uma etapa inicial do levantamento de dados para uma base de dados, que seria uma etapa textual, utilizando a língua nativa do analista. Talvez por questões de globalização ou simplesmente agilização do desenvolvimento, a maioria dos autores inicia o desenvolvimento do modelo de abstração com a diagramação de entidades e relacionamentos. Poucos são os alunos de um curso de graduação na área da computação que têm visão lógica suficiente de sistemas para conseguirem interligar as diversas entidades necessárias para o desenvolvimento do sistema. Geralmente as entidades são identificadas, porém seus atributos e relacionamentos nem sempre são claros o suficiente para que um estudante consiga visualizá-los e aplicar os conceitos embutidos no modelo relacional.

A técnica de lista diagrama não visa substituir o modelo relacional, porém visa auxiliar o entendimento do sistema para que seja criado um modelo consistente de banco de dados que suporte ao sistema que se deseja construir, o que somente poderá ser realizado com o total entendimento das regras que regem o sistema que está sendo desenvolvido.

1.2 - Objetivo

Esta pesquisa tem como objetivo analisar a viabilidade do desenvolvimento de uma ferramenta de análise textual, baseada em regras, capaz de corrigir falhas em um levantamento de necessidades que tenha como foco a criação de um sistema de informações suportado por um banco de dados relacional.

Como a técnica desenvolvida é baseada em regras, existe a possibilidade de se desenvolver um software que analise um texto escrito e indique as possíveis falhas, utilizando para isso a base teórica de tratamento de linguagem natural dentro da Inteligência Artificial.

O objetivo deste software será a análise de frases escritas para a representação da realidade dentro do modelo conceitual do sistema, observando a análise de cada frase que levará em conta os seguintes itens:

- Análise léxica dentro de cada frase;
- A análise sintática da frase individualmente;
- Redundâncias e controvérsias nas definições;
- A coerência contextual do levantamento de dados.

Esta ferramenta deverá ainda indicar possíveis erros ou dúvidas que possam afetar a integridade do sistema a ser desenvolvido dentro do modelo conceitual de dados.

1.3 - Justificativa

Na área de análise de banco de dados, assim como tantas outras, o aluno deve ser capaz de abstrair toda a necessidade do usuário que deseja um sistema de informação para poder entender e confirmar o que deve ser criado. Neste enfoque, o aluno não pode ser somente capaz de reproduzir o que foi aprendido em sala de aula, mas deve ser capaz de entender a realidade e montar a sua própria solução que, apesar de estar baseada em regras conceituais, deve ser criativa e inteligente para que possa atender de forma eficiente às necessidades de quem está solicitando um sistema de informação.

O uso da Técnica de Lista Diagrama tem apresentado bons resultados, pois normatiza o primeiro nível do modelo de abstração, o nível conceitual. A grande maioria dos autores que tratam do assunto Sistemas de Banco de Dados iniciam suas normatizações somente em nível lógico de abstração, ou até iniciam em nível conceitual, porém tratando-o de forma gráfica, através do modelo Entidade-Relacionamento, que é muito distante do conhecimento inicial de um aluno que está aprendendo a realizar levantamento de dados ainda e dificilmente consegue visualizar entidades e relacionamentos em um primeiro contato. Outro fator agravante está a falta de conhecimento de rotinas administrativas de

uma organização, onde são desenvolvidos a maioria dos sistemas de informação. Ora, se um aluno não tem um conhecimento bom sobre o assunto que necessita ser representado, ele necessita aprender simultaneamente sobre o negócio e a técnica, o que torna o aprendizado não muito produtivo. O uso da técnica de Lista-Diagrama, mais precisamente do primeiro passo da técnica, possibilita ao aluno utilizar a linguagem escrita, que é de seu conhecimento, para representar o sistema que será desenvolvido, proporcionando maior confiabilidade na técnica, que não deixa de exigir um diagrama e as tabelas relacionadas, conforme normas definidas por Peter CHEN (1976).

1.4 – Estrutura do Trabalho.

Para melhor desenvolvimento e entendimento, este trabalho está organizado da seguinte forma:

a) Introdução

Uma breve explicação sobre o trabalho a ser desenvolvido.

b) Revisão Bibliográfica.

Nesta parte serão revisados os conceitos que envolvem o desenvolvimento desta dissertação.

c) Proposta de Solução.

Análise do problema apresentado e desenvolvimento lógico de uma solução pautada nos conceitos revisados.

d) Desenvolvimento do Sistema.

Nesta parte será projetada uma solução para o problema apresentado, validando a análise anterior.

e) Apontamentos das próximas etapas.

2. - Revisão Bibliográfica

2.1 - Sistemas de Informações

Com a popularização dos computadores, através da microinformática, as empresas, cada vez mais, adquirem equipamentos de informática e, conseqüentemente, desejam que seja desenvolvido um sistema de informações que a auxilie na administração das suas atividades diárias.

Por muitas vezes a empresa busca no mercado softwares prontos que atendam às suas necessidades, porém, em virtude da complexidade e diversidade de modelos de negócios existentes, os chamados softwares “de prateleira” se apresentam ineficientes, ou seja, dificilmente atendem às necessidades das empresas sem que estas tenham que alterar a sua rotina de trabalho.

Chega-se então a um impasse: O que seria então um bom sistema de informação?

“Um bom sistema de informação é aquele de onde a empresa consegue obter respostas às suas necessidades, sem ter que abrir mão dos procedimentos de tomada de decisão ou usar recursos adicionais para complemento da informação.” BAÑOS (1997)

Como observado no mercado atual de software, podemos concluir que existe uma grande dificuldade das empresas em adquirirem software para o seu negócio, ficando sempre algo faltando ou, por muitas vezes, algo sobrando, o que traz certa insegurança e insatisfação para a organização, visto não conseguir utilizar o software na sua totalidade.

Tendo em vista estes fatos, o administrador, na maioria das vezes busca no mercado empresas ou profissionais que desenvolvam um sistema de informações baseando-se nas necessidades da sua empresa. Para este desenvolvimento, geralmente, o projetista utiliza alguma metodologia, como por exemplo:

- Análise em Cascata;
- Engenharia Reversa;
- Engenharia da Informação;
- Business Design;
- Modelagem Orientada a Objetos;
- Componentização;
- Outras metodologias.

O uso destas metodologias garante, tanto ao projetista quanto ao cliente um certo grau de segurança em relação ao sistema que está sendo criado.

Uma analogia que pode ser aplicada à etapa de análise e projeto do sistema é a da construção de um prédio, onde o engenheiro pode tentar adaptar algum projeto já desenvolvido ou criá-lo inteiramente. Nas duas situações ele tem que levantar junto ao seu cliente as suas necessidades e desejos colocá-los em um projeto e aprová-los junto ao cliente antes do início da obra. Além desta aprovação existem ainda outras aprovações, como a de segurança, que é realizada, geralmente, pelo corpo de bombeiros local, e a de conformidade, que é realizada junto ao poder executivo/legislativo do município onde a obra será realizada.

Quando falamos em informática, não existem ainda órgãos que vistoriem projetos, nem ao menos a obrigação da criação de um projeto, onde, por muitas vezes, leva o empresário que deseja um sistema de informações para a sua empresa, a aceitar um projeto sem ter visualizado o que será desenvolvido e aceitar prazos, que dificilmente são respeitados. Esta visualização é obtida, por vezes, através da aplicação de técnicas de desenvolvimento de sistemas. As técnicas devem sempre focar a organização de dados da empresa e suas atividades.

Sabe-se que a engenharia de software passa por uma crise, pois as tecnologias avançam vertiginosamente a cada dia, e, as abordagens utilizadas no desenvolvimento de sistemas continuam sem muitas modificações. Muitos desenvolvedores ainda acreditam

poder desenvolver sistemas em pouco tempo e sem um projeto. Serão analisados agora alguns dos paradigmas atuais utilizados para desenvolvimento de sistemas.

2.2 - Paradigmas de Desenvolvimento de Sistemas de Informação.

Qualquer projeto de Banco de Dados é iniciado com a coleta de requisitos para a aplicação e a sua respectiva análise, onde, em etapa seguinte deve-se desenvolver o chamado projeto conceitual do sistema. O projeto conceitual resulta em um esquema conceitual do banco de dados, que é uma descrição da estrutura do banco de dados em um alto nível de abstração, independente do sistema gerenciador de banco de dados (SGBD) que será utilizado para o desenvolvimento. A partir deste projeto conceitual é gerado o projeto lógico do sistema, que inicia com o esquema conceitual e resulta em um esquema lógico, que é uma descrição de um banco de dados em termos da classe de SGBD adotada (hierárquico, redes, relacional, orientado à objetos). Finalmente, a partir do modelo lógico, é gerado o projeto físico que tem origem no esquema lógico e resulta em um esquema físico que descreve a base de dados de acordo com um SGBD específico. (ELMASRI, 2000).

Focando em SGBD's baseados no modelo relacional, muitas são as técnicas utilizadas para o desenvolvimento de sistemas sobre base de dados. Muitos paradigmas, modelos ou técnicas são baseados na experiência do analista, o que restringe o desenvolvimento profissional de sistemas de informação a poucos. Outros são muito abertos, dificultando o entendimento ou aprendizagem por falta de métricas ou regras claras para o desenvolvimento do projeto por parte do projetista. A seguir algumas das principais técnicas de desenvolvimento são detalhadas para uma melhor percepção do assunto.

2.2.1 - Ciclo de Vida Tradicional ou em Cascata.

O ciclo de vida tradicional, ou em cascata, tem como característica a não geração de um modelo de abstração formal, não sendo utilizadas técnicas de estruturação e quase não existe a interação de usuário no desenvolvimento do sistema, sendo que ao término obtém-se

um sistema com pontos definidos pelo projetista por conhecimento ou simples experiência, desprezando, por vezes, as reais necessidades do usuário. A aplicação deste ciclo de vida tem suas atividades desenvolvidas em seqüência e não existe retorno de atividades, gerando um alto índice de manutenção. (MACHADO & ABREU, 1995).

2.2.2 - Engenharia Reversa

A seqüência de etapas para o desenvolvimento natural de um banco de dados leva em consideração a hierarquia básica do modelo clássico de abstração, ou seja, esquema conceitual, lógico e físico, porém, por muitas vezes o analista se defronta com um sistema já desenvolvido e em produção há algum tempo. Este mesmo sistema, por vezes não necessita ser reescrito. Para este sistema pode ser aplicado o processo da engenharia reversa, onde o projetista irá inverter a cadeia natural de desenvolvimento do modelo de abstração clássico, ou seja, ele deseja derivar um esquema conceitual a partir de um esquema físico. Segundo SHOVAL (1993) as razões para a aplicação deste método são:

- a) Existe um sistema antigo criado no passado que é ainda importante. Devido a evolução da tecnologia, um sistema desenvolvido no passado, e que ainda é de importância para uma organização, adquire várias deficiências, ou seja: (1) parte do código nunca é utilizado; (2) há redundância no código; (3) há inconsistência no código; (4) existem trechos de código de difícil entendimento. Como resultado dessas deficiências, existem dificuldades em manter o sistema.
- b) Existe a necessidade de mudar para uma tecnologia diferente (Sistema Operacional, Arquitetura de Processadores, Evolução de Redes, etc). Essa nova tecnologia pode gerar incompatibilidade com as ferramentas já existentes.
- c) A habilidade de expressar a estrutura de um banco de dados em um esquema onde existe maior informação semântica que o sistema implementado. Recuperar ou descobrir o domínio semântico que não é

representado explicitamente no esquema relacional (por exemplo, cardinalidade). Representar o resultado em um esquema conceitual que provê um melhor entendimento do dado com respeito ao domínio da aplicação.

- d) Identificar os componentes (e.g. relações (ELMASRI & NAVATHE, 2000)) e seus relacionamentos.
- e) Migração entre paradigmas de banco de dados. Por exemplo, de hierárquico para o relacional.
- f) Migração entre diferentes implementações de paradigma de banco de dados. Como, por exemplo, alterar o sistema gerenciador de banco de dados de uma determinada empresa de desenvolvimento de banco de dados relacional para outra que possua outro SGBD mais adequado às necessidades da organização.
- g) Obter uma documentação que não existia ou é incompatível com o sistema atual

A adoção da técnica de engenharia reversa é rica em documentação e pode ser aplicada através do uso de softwares denominados CASE (Computer Aided Systems Engineering – Engenharia de Sistemas Auxiliada pelo Computador). Muitas empresas desenvolvedoras de SGBD já provêm estas ferramentas CASE com o intuito de facilitar migrações de ambientes através da engenharia reversa, com o aproveitamento da “inteligência” já desenvolvida e, principalmente o reaproveitamento das informações já armazenadas e estruturadas em uma base de dados existente.

2.2.3 - Engenharia da Informação

A Engenharia da Informação (EI) é uma metodologia desenvolvida para ser, segundo MARTIN (1991), *"um conjunto interligado de técnicas formais de planejamento, análise, projeto e construção de sistemas de informações sobre uma organização como um todo ou em um de seus principais setores"*, ou melhor, *"é um conjunto de disciplinas automatizadas em nível de organização cuja finalidade é fornecer as informações certas às pessoas certas e na hora certa"*. Como resultado da implantação desta metodologia tem-se sistemas de informação alinhados com os objetivos da organização, compatíveis entre si, sem redundâncias desnecessárias e de baixo custo de manutenção (MARTIN, 1991).

Segundo FELICIANO NETO & FURLAN & HIGA(1988) a EI pode ser representada por uma pirâmide de quatro faces, sustentada por dados, atividades, tecnologia e pessoas, onde a face dos dados fornece a base de sustentação das informações necessárias para a sobrevivência da empresa, decisões gerenciais e financeiras, representada por atributos que compõe uma classe de dados denominada entidade. A face das atividades está voltada para os aspectos funcionais do sistema, ou seja, sustenta os processos gerenciais e as atividades de devem ser exercidas para que a organização seja eficiente na sua missão, atingindo objetivos, metas e desafios fixados. A face da tecnologia tem a preocupação da rápida evolução tecnológica que o sistema tem que possuir para atender às expectativas organizacionais e finalmente a face das pessoas tem como preocupação os recursos humanos envolvidos no desenvolvimento do projeto, seu perfil profissional e a extensão do comprometimento com cada fase da EI.

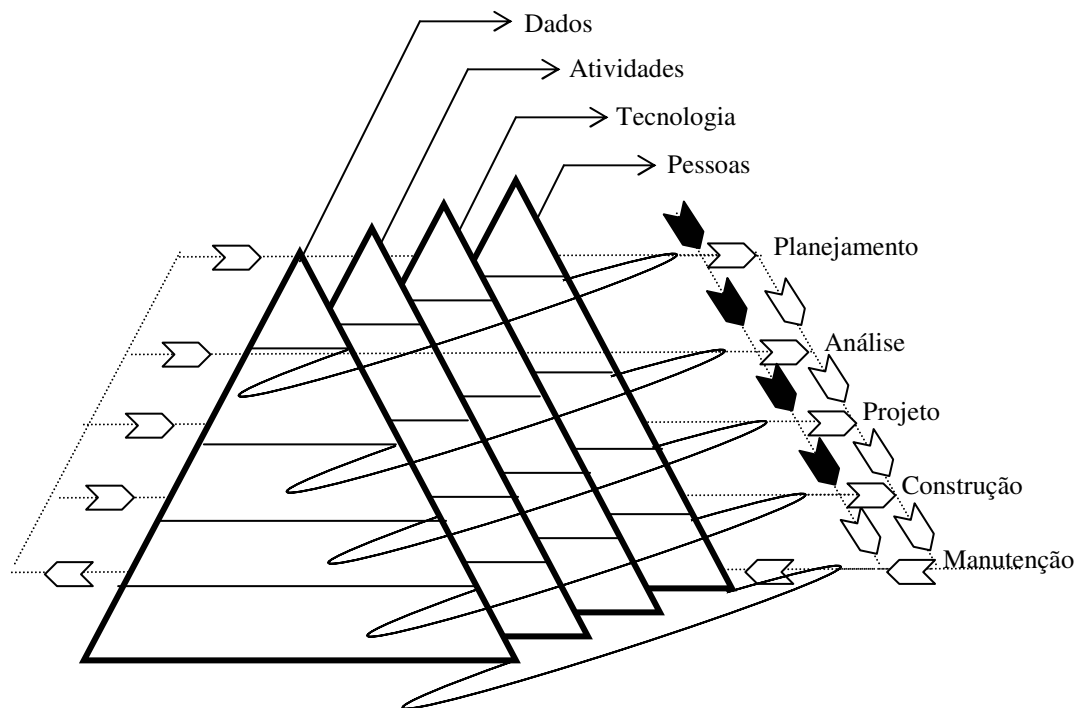


Figura 2 - Interação entre as fases da EI segundo FELICIANO NETO & FURLAN & HIGA(1988)

Como fases da Engenharia da Informação FELICIANO NETO & FURLAN & HIGA (1988) identificam cinco delas que têm como características a coesão, integridade interatividade e seqüência, que são:

- Planejamento estratégico de informações;
- Análise das áreas de negócios;
- Projeto;
- Construção;
- Manutenção.

Cada uma das fases descritas acima é responsável pela execução e incorporação de tarefas específicas. Adicionalmente, por ser implantada com o suporte de software específico, a EI gera um crescente repositório de conhecimentos sobre a organização, denominado por MARTIN (1991) de "enciclopédia". Nela ficam registrados os seus modelos de dados, modelos de processos, projetos de sistemas e dicionários (dados,

processos, variáveis). Além disso, a enciclopédia pode disponibilizar ferramentas que fazem verificações cruzadas e análises de correlação entre as variáveis armazenadas.

2.2.4 - Business Design

No início dos anos 90, a maioria das grandes corporações que possuíam sistema de informações tinha como base equipamentos de grande porte, denominados Mainframes. No Brasil duas grandes empresas detinham boa parte do mercado em seus equipamentos e softwares e, por conseguinte desenvolveram algumas filosofias de desenvolvimento, como é o caso do Business Design, desenvolvido pela BURROUGS na década de 80, e posteriormente assumido pela UNISYS, empresa que a sucedeu no mercado através da junção das empresas Sperry e Burroughs em 1986. Como diferenças básicas, o Business Design propõe que não existam as fases de análise e projeto independentes, mas sim um desenho único, onde o analista deverá efetuá-las simultaneamente, ou seja, a cada etapa de análise feita já se documenta e evolui o projeto (UNISYS, 1989). Outra técnica aplicada dentro deste conceito é o do desenvolvimento através de protótipo não descartável, denominada prototipação evolutiva, onde o usuário participa de cada fase da criação do sistema em um ciclo completo, ou seja, do desenho à implementação. Com o uso da prototipação evolutiva o uso do sistema não depende da finalização do projeto, mas a cada etapa concluída já podem ser liberados procedimentos para que o usuário utilize, agilizando a detecção de erros e obtendo maior participação durante a implementação, testes e implantação, pois não são acumuladas tarefas para uma única etapa de implantação (UNISYS, 1989).

Uma preocupação muito grande durante a modelagem do sistema é o não armazenamento de informações, mas sim de dados brutos, pois, segundo UNISYS (1989):

- Os dados brutos podem ser analisados e reconstituídos de várias maneiras;
- Possibilita auditorias no sistema;
- A carga imposta ao processador por cada transação on-line é significativamente menor se for limitada à lógica de validação somente;

- O projeto físico do banco de dados não necessita ser alterado toda vez que as necessidades de informação da organização se alteram;

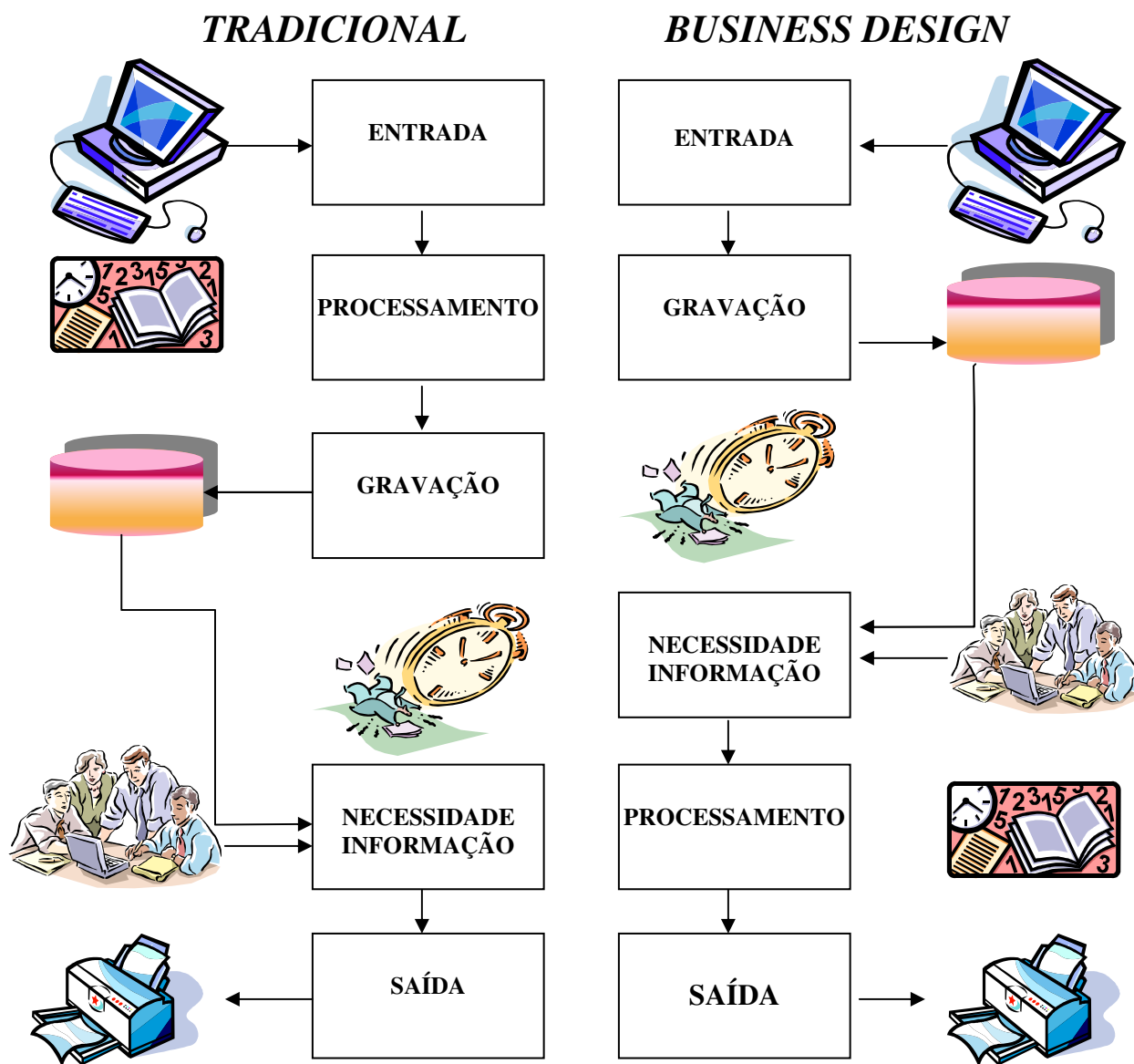


Figura 3 - Ciclo de Vida Business Design

Para o desenho do sistema utiliza-se o conceito de INTERFACE SPECIFICATION, ou simplesmente ISPECs (UNISYS, 1989), onde o analista tem que ter a preocupação de identificar a infra-estrutura do sistema de negócios da organização e as atividades desenvolvidas. A infra-estrutura de um sistema de negócios consiste dos recursos físicos do negócio tangíveis e reconhecíveis, sobre os quais as atividades ocorrem.(UNISYS, 1989).

As atividades registradas do sistema de negócios são as atividades diárias que as pessoas executam no seu trabalho. Essas atividades envolvem as necessidades de comunicar o que foi feito ao resto da organização (UNISYS, 1989).

Ainda segundo a UNISYS, para o projeto e criação de um sistema baseado em Business Design, as seguintes etapas devem ser desenvolvidas:

- a) Análise de necessidades
- b) Especificação do Sistema
- c) Geração do Sistema
- d) Execução do Sistema
- e) Controle e Auditoria do Sistema
- f) Evolução do Sistema

Esta técnica tem como princípio a Resistência do Sistema, ou seja, o sistema mantém-se evoluindo através do tempo ao invés de sofrer “remendos” no seu ciclo de vida. A estrutura básica do sistema deve ser mantida até a descontinuidade do sistema, ou melhor, até que o sistema perca a eficiência em atender a função básica para que foi projetado, dificultando a implementação do sistema.(UNISYS, 1989)

2.2.5 – Orientação a Objetos

A evolução da informática como ciência trouxe ao usuário a possibilidade de elaboração de novos projetos, que não tinham viabilidade na década de 70, quando ocorreu o desenvolvimento do modelo relacional. Muitos destes problemas, ou melhor, formatos de informação, não eram possíveis de serem tratados por sistemas computacionais, como é o caso de imagens, textos, gráficos, vídeos, sons, dentre outros. Atualmente, com a evolução dos processadores e memórias muitas informações são derivadas destes tipos de dados, não previstos no modelo relacional original e os tipos previstos anteriormente também evoluíram, como, por exemplo, as necessidades de modelagem de um sistema financeiro tradicional hoje já exigem um tratamento de dados de imagens e hipertexto. O Modelo de dados orientado a objeto foi proposto para atender às exigências dessas novas aplicações.(SILBERSCHATZ & KORTH & SUDARSHAN, 2000).

Segundo SILBERSCHATZ & KORTH & SUDARSHAN (2000), *“O modelo de dados orientado a objeto é uma adaptação a sistemas de banco de dados do paradigma de programação orientada a objeto”*.

Segundo ULLMAN & WIDOW (1997), dentro do projeto orientado a objeto, o mundo a ser modelado é imaginado como se fosse composto por objetos, que é comparado, assim como em SILBERSCHATZ & KORTH & SUDARSHAN (2000) como um correspondente superficial de uma entidade no modelo E-R. Assume-se que cada objeto possui apenas um identificador (OID) que o distingue dos outros objetos. Para efeito de organização, os objetos são agrupados em classes de objetos que possuem propriedades similares.

Ainda segundo ULLMAN & WIDOW (1997), para o desenvolvimento de banco de dados orientado a objeto, deve-se utilizar uma Linguagem de Definição de Objetos, denominada simplesmente de ODL (Object Definition Language). A ODL é uma linguagem para especificação do esquema ou estrutura do banco de dados, ou melhor, é uma linguagem que permite a definição das propriedades dos objetos.

Como propriedades, ULLMAN & WIDOW (1997) definem que o projeto de uma classe pode possuir três tipos, que são:

1. ATRIBUTOS;
2. RELACIONAMENTOS;
3. MÉTODOS.

Os atributos são o tipo de propriedade mais simples, que descrevem um aspecto de um objeto, ou melhor, a sua composição de dados, como por exemplo, um objeto PESSOA pode ser composto por um atributo NOME que é do tipo CHARACTER e recebe como valor o nome da pessoa.(ULLMAN & WIDOW, 1997).

Os relacionamentos são conexões entre objetos de classes distintas, ou de mesma classe, necessárias para compor informações. As relações são tratadas de forma derivada do modelo E-R, tendo como maior implementação o conceito de herança, que deriva do conceito de especialização do modelo E-R. A Herança é definida, segundo SILBERSCHATZ & KORTH & SUDARSHAN (2000) como uma representação de uma subclasse de uma classe “mãe” onde todas as propriedades são herdadas, acrescentando as novas propriedades da classe “filha”.

Já os métodos são funções que podem ser aplicadas nos objetos de uma classe(ULLMAN & WIDOW, 1997), ou melhor, segundo FURLAN (1998), são as operações que podem ocorrer com o objeto de uma classe designando-lhe um comportamento.

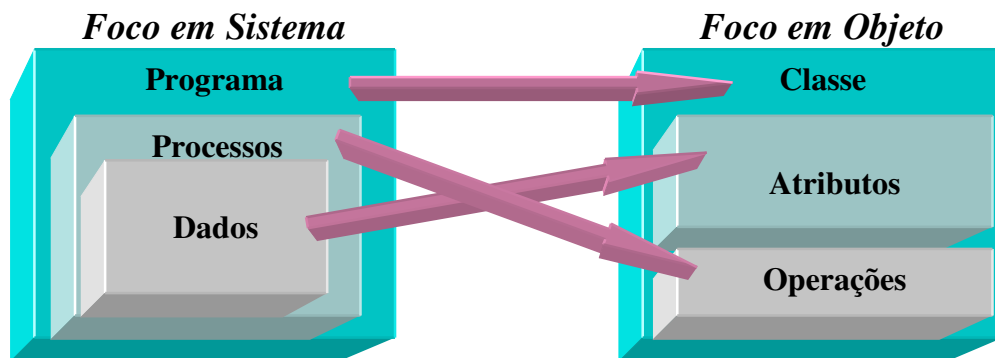


Figura 4 - Enfoque baseado em Sistema X Enfoque baseado em Objeto, segundo FURLAN (1998)

Com relação ao mapeamento no banco de dados existem, segundo SILBERSCHATZ & KORTH & SUDARSHAN (2000), duas abordagens que podem ser adotadas para se criar um banco de dados orientado a objeto: pode-se adicionar os conceitos de orientação a objeto a linguagens de banco de dados existentes, como é o caso dos bancos de dados relacionais estendidos, ou pode-se estender linguagens orientadas a objeto para tratar de banco de dados pela adição do conceitos.

Os tradicionais bancos de dados relacionais de mercado estão se adaptando aos conceitos de orientação à objetos, criando os chamados banco de dados objeto-relacional (SILBERSCHATZ & KORTH & SUDARSHAN, 2000), porém, muitos fornecedores de banco de dados, segundo FURLAN (1998), já têm seus produtos adaptados ao paradigma orientado a objeto, criando alguns software já integralmente orientado a objetos, denominados Sistema Gerenciador de Banco de Dados Objeto (ODBMS), que são o Jasmine da Computer Associates, o O2 da O2, o Object Store da Object Design, o Objectivity/DB da Objectivity e o Versant ODBMS da Versant.

Uma grande dificuldade no uso de banco de dados orientado a objeto está na falta de padronização oficial para criação de banco de objetos, deixando a cargo de cada fornecedor o foco dado a base de objetos, além da forte base instalada de bancos de dados relacionais (SGBD) que tendem a migrar para os sistema híbridos antes de mudanças radicais nas suas rotinas administrativas. (FURLAN, 1998).

2.3 - Sistema Objeto de Estudo

O projeto de um sistema de informações que tem como alicerce uma base de dados é desenvolvido respeitando-se a realidade que se deseja gerir, ou seja, que tem como finalidade a organização de dados comerciais, pessoais, científicos, dentre outros.

Para efeito de organização, quando houver referência da realidade que se está estudando será adotada a terminologia Sistema Objeto de Estudos, ou simplesmente SOE. Pode-se dizer que o SOE está inserido em um modelo maior, denominado Modelo de Abstração, que tem como finalidade sistematizar o processo de desenvolvimento de uma base de dados eficiente.

O modelo de abstração é dividido basicamente em quatro camadas, iniciando-se sempre do nível mais próximo ao usuário e concluindo-o no nível mais baixo do sistema, ou seja, próximo à linguagem de programação.

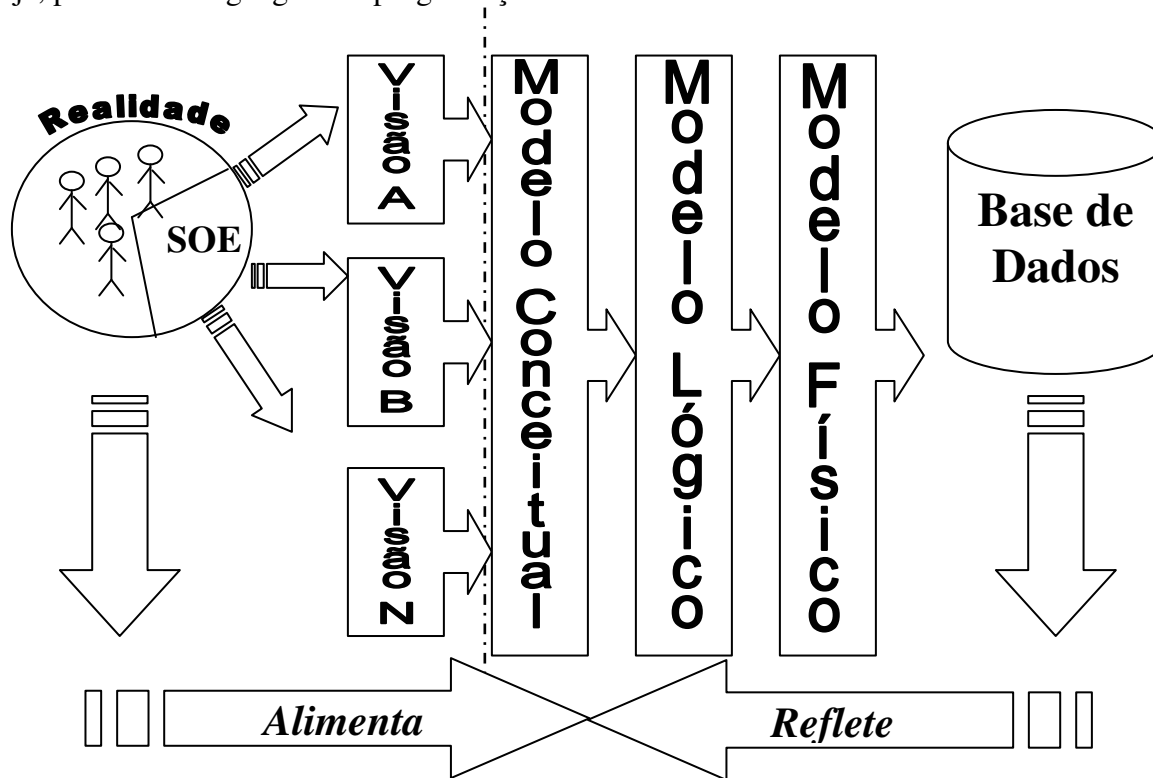


Figura 5 - Modelo de Abstração a partir do Sistema Objeto de Estudos segundo FELICIANO NETO & FURLAN & HIGA(1988)

Alguns autores como MACHADO & ABREU (1995) indicam a visão como um nível superior ao modelo de abstração, não estando incluído como uma etapa do desenvolvimento do projeto, porém outros autores como SILBERCHATZ & KORTH & SUDARSHAN (2000, p. 4 e 5) incorporam o nível de visão ao modelo de abstração, sintetizando-o ao modelo conceitual.

Em geral, como primeiro nível, o projetista deve analisar quais são as visões necessárias que esta base de dados deverá atender. Entende-se por visão toda a forma diferenciada de como os dados são apresentados. Toda visão é baseada nas necessidades de cada usuário, sendo que o projetista não poderá descuidar da integração dos dados e de sua integridade. Este nível pode ser interpretado também como nível conceitual, onde são explicitados todos os conceitos que envolvem os dados para o SOE em questão, devendo ser projetado em uma linguagem próxima à linguagem natural, quando possível utilizando a própria, pois deve ser avaliado juntamente ao(s) usuário(s) conhecedor do SOE.

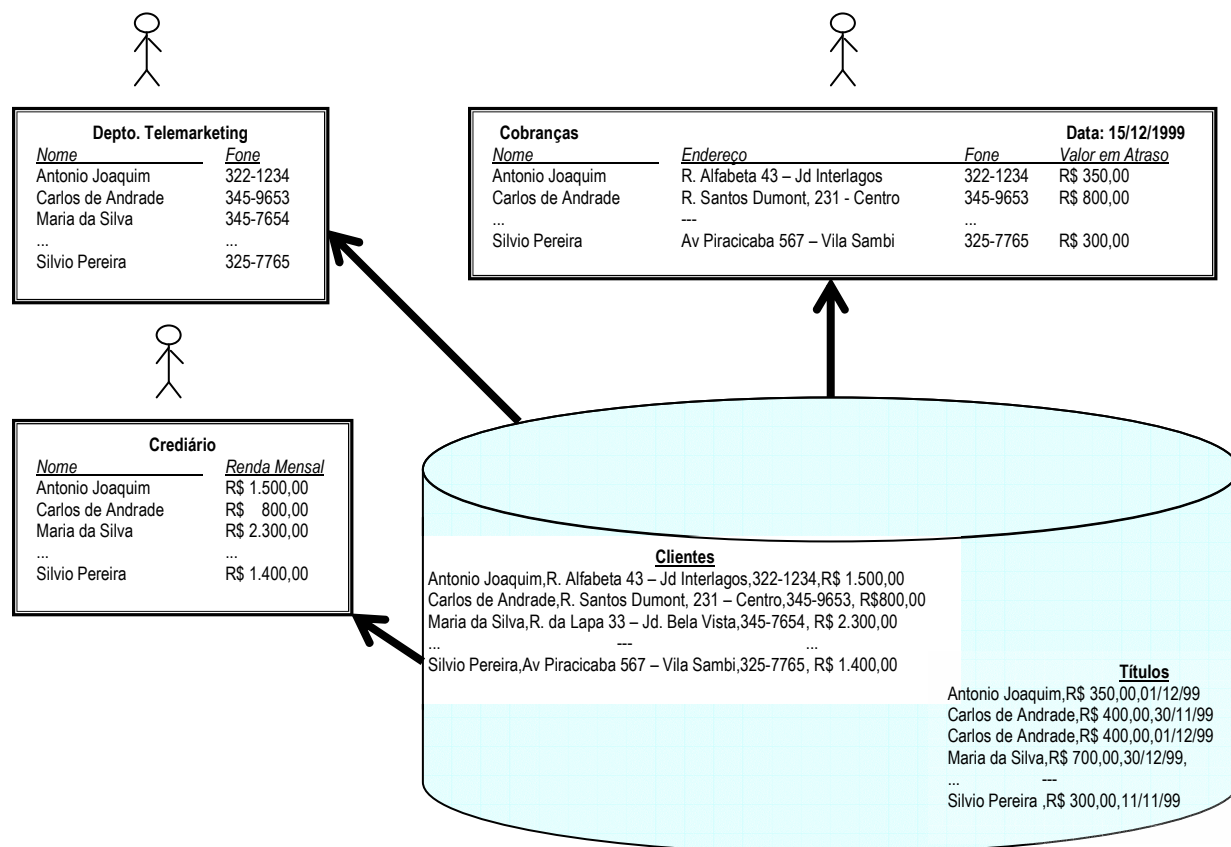


Figura 6 - Representação das Visões do SOE

O nível lógico é um nível que tem como função demonstrar os relacionamentos existentes entre os dados. Como representação é utilizado como padrão o Diagrama Entidade Relacionamento, que demonstra de forma gráfica os agrupamentos dos dados e seus relacionamentos. Este nível já se afasta um pouco da compreensão do usuário por tratar os dados de forma lógica, respeitando algumas regras de integridade que irão garantir que a informação gerada por esta base é confiável. O Diagrama Entidade Relacionamento, ou DER, é um gráfico de interpretação simples, porém é dificultado em virtude de uma falta de

padronização no seu uso, tendo diversas variações aplicadas por autores diferentes, porém a mais tradicional é exemplificada abaixo.

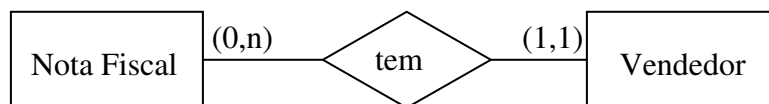


Figura 7 - Representação de modelo Lógico segundo FELICIANO NETO & FURLAN & HIGA(1988)

Já o nível físico é o que mais se aproxima da estrutura da base de dados, onde são representadas as tabelas que comporão a base, identificando as chaves (primárias e estrangeiras), afastando-se, portanto da linguagem natural e, conseqüentemente, do entendimento do usuário leigo. Esta etapa pode ser representada através do uso de tuplas ou de esquema de tabelas, sendo que a maioria da literatura consultada está optando por utilizar o formato de esquema visto este ser muito parecido com a linguagem SQL, adotada pela ampla maioria dos Sistemas Gerenciadores de Banco de Dados (SGBD) utilizados no mercado. (ULLMAN & WIDOW (1997))

VENDEDOR(Cód_Vendedor, Nome, Salário)

NOTAFISCAL (NumNota,DtVenda, CPF, Cód_Vendedor,...)

Cód_Vendedor referencia Vendedor

O Modelo de Abstração representa ainda a criação efetiva da base de dados, que pode ser realizada através do uso de softwares assistentes ou de linguagem de programação, geralmente SQL. Após a criação da base de dados, esta deve ser alimentada pelo usuário, através do Sistema de Informações que deve ter sido desenvolvido para esta finalidade e a base de dados deve propiciar a este mesmo sistema uma imagem fiel da situação do SOE, através do processamento dos dados residentes na base que gerarão as informações, que serão visualizadas através de consultas em terminais ou relatórios impressos.

2.4 - Integridade de Dados e Anomalias

Ao projetar a base de dados o analista tem por obrigação garantir a integridade da base de dados evitando a ocorrência de anomalias nos dados (ELMASRI & NAVATHE, 2000). As anomalias são ocorrências que afetarão as saídas da base de dados, ou melhor, as consultas e relatórios, colocando em dúvida a veracidade da informação. As anomalias podem ser divididas em anomalias controladas e anomalias não controladas, geralmente oriundas de falhas no projeto.

As anomalias controladas são anomalias criadas pelo projetista para satisfazer a alguma necessidade do usuário que não foi respondida pelo modelo criado, ou até, satisfazer necessidades de performance da base de dados em virtude do volume de dados. Neste caso o analista tem que ter uma documentação muito clara sobre o objetivo da anomalia para facilitar futuras manutenções na base.(HEUSER, 1999).

As anomalias não controladas são perigosas para a integridade do sistema, pois fica a cargo do usuário o seu controle, não havendo nenhum processo automatizado que a regule. Este tipo de anomalia, caso faça parte do projeto, deve ser documentada tanto na documentação do projeto quanto na documentação do usuário. (HEUSER, 1999).

As anomalias ainda são divididas em tipos, a saber: anomalia de atualização, anomalia de inclusão de dados, de alteração de dados e anomalia de exclusão de dados.

2.4.1 – Anomalias de Inclusão de Dados

Uma anomalia de inclusão ocorre sempre que uma base de dados não normalizada recebe um registro que contenha algum dado que contradiga algum dos conceitos lógicos utilizados pela organização no gerenciamento do SOE. No exemplo a seguir, se deseja incluir um novo registro para o pedido 357, na data de 21/03/92, com a quantidade de 192 itens, o CGC 345832432 e o nome do cliente Editora Formax. Observando que já existe um cliente com o mesmo número de identificação do cliente, neste caso o CGC, pode-se

concluir que mudou a razão social da Ferragens Marabu e agora será necessário acertar todos os registros da Ferragens Marabu, ou foi cadastrado o CGC errado.

Pode-se ainda tentar incluir um registro com os campos Pedido, Data e Quant. em branco, preenchendo apenas o CGC 345832432 e o nome do cliente como Editora Campos. Neste caso haveria o cadastramento de uma Editora sem pedido.

Pedido	Data/Pedido	Quant.	CGC/Cliente	Nome do Cliente
353	18/03/95	100	345832432	Ferragens Marabu
354	19/03/95	150	223384471	Papelaria Barela
355	20/03/95	102	983457323	Livraria Brasil
356	21/03/95	192	223384471	Papelaria Barela

Figura 8 - Base de dados para exemplo de anomalia de inclusão

2.4.2 – Anomalias de Alteração de Dados.

Uma anomalia de alteração ocorre quando algum dos dados da base sofre manutenção, passando a possuir outro valor. Esta anomalia ocorre na base de dados quando o analista não tem a preocupação de validar as alterações realizadas ou quando este despreza os recursos existentes nas bases de dados relacionais. Para exemplificar pode-se alterar o nome do cliente do pedido 353 a seguir, passando de Ferragens Marabu para Ferragens Marabá. Neste caso pode-se concluir que o cliente teve uma alteração no seu nome e, portanto devem-se atualizar todos os pedidos realizados por este cliente. Ainda, como outro exemplo, pode-se realizar a alteração do CGC no pedido 356, onde o mesmo cliente ficaria com dois CGC constantes na base de dados.

Pedido	Data/Pedido	Quant.	CGC/Cliente	Nome do Cliente
353	18/03/95	100	345832432	Ferragens Marabu
354	19/03/95	150	223384471	Papelaria Barela
355	20/03/95	102	983457323	Livraria Brasil
356	21/03/95	192	223384471	Papelaria Barela

Figura 9 - Base de dados para exemplo de anomalia de alteração.

2.4.3 – Anomalias de Exclusão de Dados.

A anomalia de exclusão ocorre em virtude da eliminação da informação da base em virtude da deleção de algum registro. Esta perda de informação é muito perigosa, pois dificilmente poderá ser recomposta e esta falta de informação pode levar o usuário a alguma decisão errada por falta de dados. Para exemplificar pode-se eliminar o pedido 355, onde, conseqüentemente perder-se-ia as informações sobre o cliente Livraria Brasil

Pedido	Data/Pedido	Quant.	CGC/Cliente	Nome do Cliente
353	18/03/95	100	345832432	Ferragens Marabu
354	19/03/95	150	223384471	Papelaria Barela
355	20/03/95	102	983457323	Livraria Brasil
356	21/03/95	192	223384471	Papelaria Barela

Figura 10 - Base de dados para exemplo de anomalia de exclusão

A solução para que estas anomalias não ocorram é encontrada através da aplicação de regras que produzam uma base de dados atômica e livre de redundâncias. Estas regras, denominadas formas normais, se baseiam nas dependências funcionais existentes entre os atributos definidos na base de dados.

2.5 - Dependências Funcionais

Um dos conceitos mais importante para o desenvolvimento de sistemas de bases de dados relacionais é o de Dependência Funcional, pois através deste conceito podem-se identificar as chaves das relações que estão envolvidas na base, assim como validá-las, através da aplicação de regras, que serão explicitadas a seguir.(ULLMAN, 1988)

Segundo ELMASRI & NAVATHE (2000, p.476), *uma dependência funcional é uma restrição entre dois conjuntos de atributos da base de dados*¹.

Supondo que R seja uma relação com esquema R(X,Y,Z), pode-se buscar na realidade que sempre que X receber um determinado conteúdo, Y assumirá um conteúdo invariável, até que X assuma um novo valor. Pode-se dizer então que Y somente variará de valor através de X, ou ainda, que Y tem dependência funcional de X e, conseqüentemente, X determina Y, que é representado como $X \rightarrow Y$, observando que para cada X existirá apenas um valor de Y.

Uma característica importante nesta dependência funcional é que pode-se dizer que $X \rightarrow Y$, porém não se pode afirmar que $Y \rightarrow X$, salvo haver alguma regra conceitual que permita esta afirmação.(ULMANN, 1988)

O uso das dependências funcionais permite que sejam determinadas as superchaves, as chaves candidatas e, conseqüentemente, a chave primária de cada relação através da clausura² de conjunto de dependências funcionais, que é o conjunto de todas as dependências funcionais logicamente implícitas derivadas de um conjunto inicial de dependências funcionais definidas no SOE.(SILBERSCHATZ & KORTH & SUDARSHAN, 2000).

Para o uso do fecho das dependências funcionais de forma eficiente existem 3 axiomas definidos em ULLMANN & WIDOW (apud Armstrong, 1997, p. 135) e

¹ Tradução livre feita pelo autor do trabalho

² Nota de Tradução – O termo clausura é adotado por SILBERSCHATZ & KORTH & SUDARSHAN a partir da tradução da terceira edição(2000) do livro. Até então o termo utilizado era fecho, o que ainda é adotado por muitos outros autores.

SILBERSCHATZ & KORTH & SUDARSHAN (apud Armstrong, 2000, p. 204/205) que são:

- **Reflexividade** – Se $\{B_1, B_2, \dots, B_m\} \subseteq \{A_1, A_2, \dots, A_N\}$ então $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_m$. Estas são chamadas dependências triviais.
- **Incremento** – Se $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M$, então $A_1 A_2 \dots A_N C_1 C_2 \dots C_K \rightarrow B_1 B_2 \dots B_M C_1 C_2 \dots C_K$ para qualquer conjunto de atributos $C_1 C_2 \dots C_K$.
- **Transitividade** – Se $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M$ e $B_1 B_2 \dots B_M \rightarrow C_1 C_2 \dots C_K$ então $A_1 A_2 \dots A_N \rightarrow C_1 C_2 \dots C_K$.

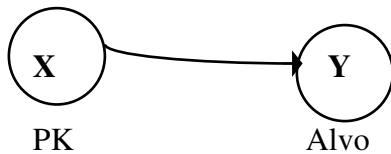
Segundo SILBERSCHATZ & KORTH & SUDARSHAN (2000) “estas regras são sólidas, porque elas não geram nenhuma dependência funcional incorreta.”.

Apesar destas regras serem completas é necessário definir novas regras, derivadas destas primeiras, que agilizam a determinação da clausura de atributos, que são: (SILBERSCHATZ & KORTH & SUDARSHAN, 2000)

- **União** – Se $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M$ e $A_1 A_2 \dots A_N \rightarrow C_1 C_2 \dots C_K$ então $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M C_1 C_2 \dots C_K$.
- **Decomposição** – Se $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M C_1 C_2 \dots C_K$ então $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M$ e $A_1 A_2 \dots A_N \rightarrow C_1 C_2 \dots C_K$.
- **Pseudotransitividade** – Se $A_1 A_2 \dots A_N \rightarrow B_1 B_2 \dots B_M$ e $B_1 B_2 \dots B_M C_1 C_2 \dots C_K \rightarrow D_1 D_2 \dots D_w$ então $A_1 A_2 \dots A_N C_1 C_2 \dots C_K \rightarrow D_1 D_2 \dots D_w \rightarrow B_1 B_2 \dots B_M C_1 C_2 \dots C_K$.

2.5.1 - Dependências Funcionais Simples. (DS).

Dizemos que existe uma DS entre os Atr **X**, **Y** quando um 'valor' de **X** está associado a *um e somente um* valor de **Y** (simples ou compostos) e representamos:



O círculo de **X** é denominado PK (Primary Key - chave primária) de **Y** e o círculo de **Y** é Alvo de **X**.

Estas DS podem assumir duas funções para qualquer valor de **X**:

- a).- sempre está associado a um valor de **Y**.
- b).- pode estar associado a um ou nenhum valor de **Y**

Ao **Y** do primeiro tipo se denomina atributo Obrigatório e ao do segundo tipo atributo facultativo.

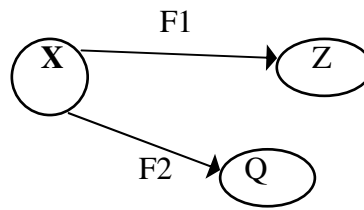
Segundo BAÑOS(1997), para distinguir pela descrição uma dependência simples Obrigatória de uma Facultativa, deve-se ser cuidadoso ao escolher os qualificadores na descrição do sistema, utilizando, por exemplo:

- .- quando houver um.
- .- sempre um.
- .- pode indicar um.
- .- etc

Em um Diagrama de Dependências, coloca-se uma letra **F** seguida de um número consecutivo que enumerará todas as *DS Facultativas* que existam em **S**.



Obrigatória



Facultativas.

Ainda no exemplo, conclui-se que os valores de **X**:

- formam sempre uma chave.
- não podem ser nulos.
- não se podem repetir.
- não são atualizáveis.

Para os valores de **Y** observamos o seguinte:

- podem não existir (caso dos facultativos).
- podem ser nulos (devem ser evitados).
- podem repetir.
- são atualizáveis.

2.5.2 - Dependências Multivaloradas (DMV)

Diz-se que existe uma dependência multivalorada entre os atributos **X** e **Y** quando para cada valor de **X** está relacionado um ou mais valores de **Y**, que pode ser representado como:

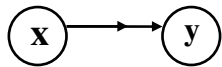


Figura 11 – Representação de Dependência Multivalorada

Uma DMV entre **X** e **Y** pode ser interpretada como, *Uma combinação de DS entre X e Y.*(BAÑOS, 1997)

Exemplo:

Cada pessoa pode estar associada a duas ou mais Profissões e/ou dominar dois ou mais Idiomas. Isto será demonstrado:

1) ID-Pessoa $\rightarrow\rightarrow$ Cod-Profissão.

2) ID-Pessoa $\rightarrow\rightarrow$ Cod-Idioma.

Para os valores **X** e **Y** pode-se dizer que:

- *ambos formam uma chave.*
- *não podem ser nulos.*
- *não pode repetir nesta combinação.*
- *não são atualizáveis.*

2.6 - Técnica Lista Diagrama

Para o desenvolvimento de projetos de bases de dados muitas são as técnicas existentes que, baseando-se em metodologias já sedimentadas, possibilitam a criação física de sistemas com grande grau de eficiência e integridade, garantindo ao projetista a segurança necessária para repassar seu projeto a fase de programação e produção.

O enfoque Lista-Diagrama não constitui uma nova Metodologia de Projeto, nem está incluído como parte de alguma Metodologia de Análise e Projeto. Por tal razão não conta com ferramentas explícitas para detectar e corrigir deficiências estruturais nos fluxos informativos de um Sistema de Informação, porém é possível detectar e corrigir incoerências e redundâncias que, de certa forma, podem ser consideradas regras corretivas de retroalimentação.(BAÑOS, 1997)

2.6.1 - Passos para projeto através de Lista Diagrama

O nome lista-diagrama origina-se pelo fato de que esta técnica está baseada em três níveis básicos, definidos pelo autor como “passos”. No primeiro passo, denominado Lista de Dependências, o projetista deverá ser capaz de abstrair textualmente as necessidades do sistema que está sendo projetado. No segundo passo, denominado Diagrama de Dependências, a preocupação está na representação gráfica das dependências funcionais existentes no sistema e, finalmente, no terceiro passo o projetista deverá interpretar o segundo passo e montar as tabelas necessárias ao sistema, representando seus atributos, chaves primária e estrangeira, além dos relacionamentos. Como o núcleo da técnica está localizado nos dois primeiros passos (Lista de Dependências e Diagrama de Dependências) e, tendo em vista que todos estes passos seguem regras rígidas e claras, pode-se imaginar a

criação de ferramentas que auxiliem o projetista na interpretação e apontamento de falhas, o que será discutido posteriormente.

O gráfico a seguir demonstra a inter-relação existente entre os três passos definidos na técnica:

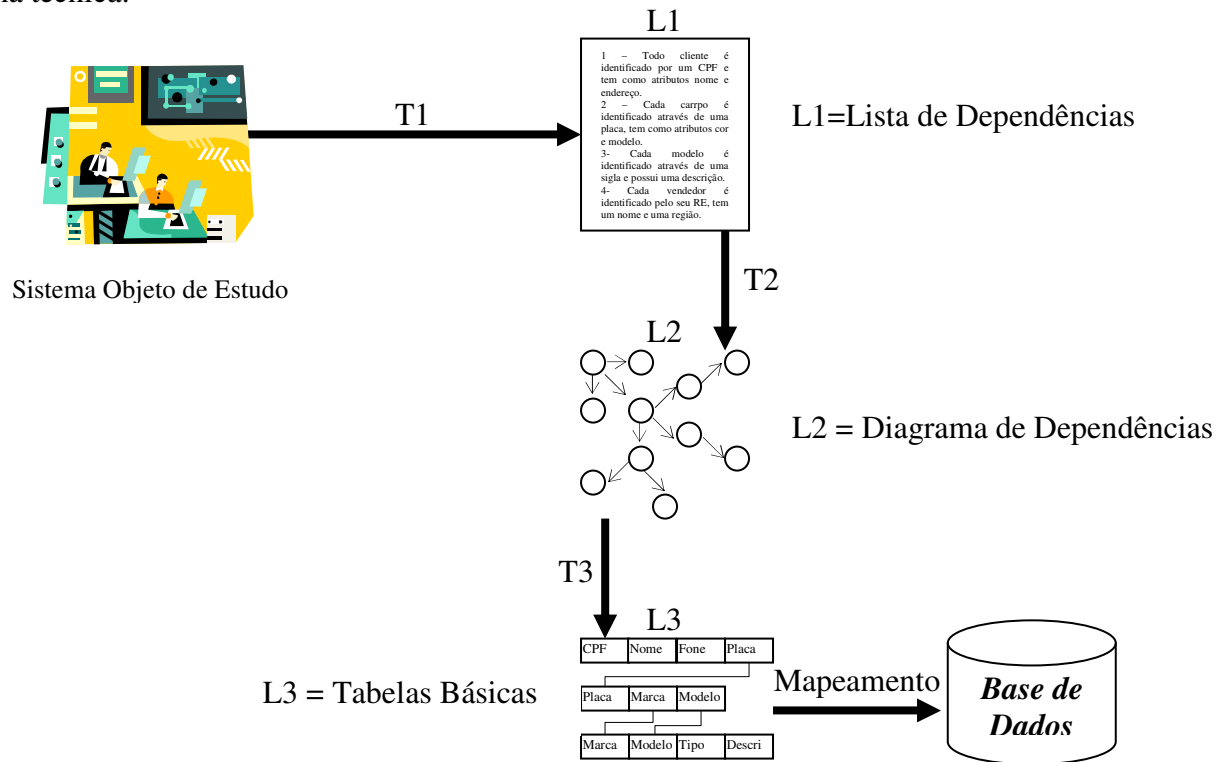


Figura 12 - Modelo de Abstração em Lista Diagrama segundo Baños (1997)

No modelo representado pela Fig. 12, pode-se dizer que a realidade onde está o Sistema Objeto de Estudo é observada e traduzida, com a aplicação de regras T1, resultando em L1, que é uma representação abstrata em linguagem textual denominada Lista de Dependências. O objetivo deste primeiro passo é capacitar o projetista a escrever as regras que envolvem o seu sistema para, juntamente com o(s) usuário(s) validar o entendimento. Posteriormente, após análise criteriosa e avaliação de L1, o projetista deverá aplicar as regras de tradução T2, com o objetivo de criar um diagrama com todos os atributos pertinentes ao SOE em questão, representando seus relacionamentos e cardinalidade. Este passo, denominado Diagrama de Dependências, gera uma representação em L2 que possibilita uma visão global do sistema e suas relações, além de obrigar o conhecimento

total das regras do negócio por parte do projetista, sem as quais não é possível a criação do diagrama sem violar alguma das regras T2. Finalmente, já no terceiro passo, o projetista deverá aplicar as regras T3 com o objetivo de criar uma representação abstrata das tabelas básicas que existem no sistema e seus relacionamentos. Nesta etapa são denominadas tabelas básicas, o que, segundo [Date/1991], são tabelas autônomas e que recebem um nome, além de plenamente normalizadas. Estas tabelas ainda podem sofrer um processo de desnormalização por parte do projetista baseado em necessidades performáticas, onde podem ser geradas dependências funcionais parciais ou transitivas e, conseqüentemente redundâncias que deverão ser gerenciadas (ELMASRI & NAVATHE, 2000, p.554). Após a criação das tabelas básicas, o último passo será então criar a base de dados física, utilizando para um esquema ou dicionário de dados pertinente ao software que estiver sendo utilizado.

2.6.2 - Atividades associadas ao 1º Passo.

O primeiro Passo tem como objetivo a criação de um modelo abstrato textual do Sistema Objeto de Estudo, onde o projetista deverá obter o conhecimento do sistema através de entrevistas com pessoas envolvidas nos processos que conheçam muito bem os detalhes e as regras que são pertinentes aos processos que estejam sendo informatizados, assim como todas as suas características e saídas que serão requeridas do sistema de informações. Neste ponto então este projetista estará capacitado a expressar o sistema (S) em lista de dependências, ou melhor, L1.

Das entrevistas com os usuários deve ser obtido:

1- Uma lista completa de todas as Entidades de interesse que formam o Sistema Objeto de Estudo, ordenando por importância relativa.(Entidades Regulares e Fracas)

2- Uma lista completa de todas as Associações de interesse em que participem as Entidades anteriores.

3- Uma lista completa de todas os atributos que envolvem as Entidades listadas em 1 e 2.

Vale ressaltar ainda que em todos os casos deve ficar bem clara a natureza do vínculo entre as Entidades identificadas e seus atributos, assim como todos os vínculos entre as entidades e as restrições pertinentes.

A “forma” que terá esta descrição é a primeira característica distintiva do procedimento.

Um dos segredos mais importantes deste procedimento está nas restrições da linguagem para construção desta **LISTA (T1)**.

A Lista de Dependências (**LD**) é formada por descrições, em Linguagem Natural, das dependências que existem entre os atributos que representam a Entidade e outros que representam às suas propriedades assim como entre os atributos que representam às Entidades e outros do mesmo tipo. (BAÑOS, 1997).

Pode-se dizer que um conjunto de atributos pode representar Entidades:

- .- Regulares.
- .- Fracas.
- .- Agregadas.
- .- Generalizadas.
- .- Especializadas.

Assim como as Propriedades de qualquer uma delas, quando houver.

Neste procedimento o projetista deverá possuir sólidos conhecimentos sobre o conceito de Dependência Funcional Simples (DS), além das Dependências Multivaloradas (DMV).

2.6.3 - T1.-Regras para construir Listas de Dependências.

As regras definidas em T1 têm como objetivo nortear o projetista no desenvolvimento de um modelo textual do projeto que está sendo desenvolvido. Tendo em vista uma normalização de linguagem, algumas regras estão definidas baseadas nos conceitos de desenvolvimento de sistema de Banco de Dados, conceitos lingüísticos, além de regras necessárias que possibilitem a clara definição das entidades, seus relacionamentos, atributos identificadores e atributos componentes.

A primeira regra de T1 diz respeito às dependências funcionais, onde existe a proibição na mistura entre dependências funcionais simples e dependências multivaloradas em uma mesma frase da definição. Esta prática geralmente leva o projetista à violação da segunda, quarta e quinta formas normais, visto que este pode não obter uma visão clara das dependências funcionais envolvidas no sistema objeto de estudo. Definiu-se então o seguinte:

*1-Cada membro da Lista de Dependências(LD) deve ser usado para descrever uma e somente **DMV**, ou várias **DS**, desde que estas **DS** estejam fazendo referência a um mesmo sujeito ou Entidade. [Banões/1997]*

Esta regra denota claramente que não podem ser colocadas na mesma frase Dependências Simples(DS) e Dependências Multivaloradas (DMV). Também deve ser observado o sujeito da frase, onde somente podem ser referenciados na mesma frase atributos relativos a um só sujeito ou entidade quando em dependência funcional simples. Esta regra ainda proíbe o uso de mais de uma dependência funcional multivalorada(DMV) na mesma frase para que fiquem bem definidas as entidades de relacionamento, originárias no DER de relacionamentos com cardinalidade N:N.

Na segunda regra foi definido que todas as frases da lista de dependências devem ser identificadas com um número, pois posteriormente serão utilizados em referências no momento da criação do diagrama de dependências – segundo passo – e, em caso de erros encontrados através deste diagrama, o projetista terá grande facilidade em atualizar seu

projeto de sistema, mantendo assim a coerência do Projeto do Sistema. Definiu-se assim então a segunda regra:

2-Cada membro da LD deve ser identificado com um número para resolver problemas de referência cruzada entre a LISTA e seu DIAGRAMA.(BAÑOS, 1997)

Outro problema geralmente encontrado em definições de sistemas está relacionado com a nomenclatura utilizada nos atributos e/ou entidades. Por muitas vezes o projetista utiliza sinônimos para referenciar um mesmo atributo. Esta falha pode comprometer o entendimento do sistema, visto que uma mesma entidade ou atributo pode ser considerado de forma distinta quando referenciado mais de uma vez através de sinônimos. Isso evita que em uma mesma definição o projetista faça referência à, por exemplo, um cliente, um correntista e ao dono de uma conta corrente em um sistema financeiro voltado ao gerenciamento bancário. Para que no projeto baseado em lista diagrama esta falha não ocorra, a seguinte regra foi definida:

3-Os elementos que compõem um membro da LD serão identificados de maneira única em toda a LD. Não podem ser usados sinônimos para referenciar o mesmo elemento.(BAÑOS, 1997)

Outra preocupação muito importante diz respeito à forma como as frases são escritas. É muito comum o projetista escrever uma frase dentro de um projeto com o sujeito oculto ou subentendido. Neste caso, para evitar uma interpretação dúbia, o projetista deverá, obrigatoriamente utilizar a seguinte regra:

4-Os membros da LD serão redigidos de modo que esteja claro quem é o sujeito da dependência.(BAÑOS, 1997).

Como exigência definiu-se ainda a seguinte estrutura:

.- <sujeito><verbo><predicado> onde:

- Sujeito.- será (s) atributo(s) representante da Entidade Determinante.

-Verbo.- definirá a dependência entre o sujeito e suas propriedades ou uma associação do sujeito com outra entidade, no caso da frase estar expressando um relacionamento entre entidades.

.-Predicado.- define a Entidade ou propriedade(s) associada através dos atributos que a compõe.

A cardinalidade das dependências também deve ser explicitada no projeto visto serem de suma importância para entendimento e definição dos relacionamentos entre as entidades. Toda cardinalidade tem que ser bem analisada e sempre questionada sobre a sua veracidade, pois, caso exista uma má definição de cardinalidade, todo o projeto pode ser comprometido pela ineficiência do sistema em gerar informações, ou impactar negativamente durante a programação do sistema exigindo da linguagem maior quantidade de algoritmos de controle do que na realidade necessitaria para que seja controlada a redundância que esta falha gera. Baseando-se nesta necessidade, foi então definida a seguinte regra:

5-A natureza das dependências (DS ou DMV) descritas tem que estar bem clara. Isso se garante escolhendo adequadamente as combinações verbo-artigo do predicado utilizado.(BAÑOS, 1997)

Como exemplo podem ser utilizadas as seguintes combinações:

Tem um, está associado a um, possui um, etc;

quando mais que um, adquire vários, mais do que um , ao menos dois, etc.

A atomicidade de atributos também é muito importante na definição de um projeto de banco de dados, onde atômico significa que cada valor em um domínio é indivisível, para que sejam atendidos os requisitos do modelo relacional (ELMASRI & NAVATHE, 2000), onde se definiu então a seguinte regra:

6-Cada atributo representante de Entidade ou de propriedade de Entidade, tem que assumir valores atômicos para conseguir de maneira imediata as condições de Codd para a 1FN (BAÑOS, 1997).

Para um bom entendimento do encadeamento do sistema é interessante que as frases escritas no projeto estejam ordenadas por ordem de importância, onde a frase mais importante sempre será a que descreve a principal atividade relativa ao sistema que está sendo projetado. Esta ordenação é muito subjetiva, porém serve como base lógica para que sejam validadas as relações existentes entre os atributos descritos, podendo, portanto haver divergência de interpretação de importância entre projetistas diferentes, o que não afetará o sistema a ser desenvolvido pois esta regra serve apenas como referencial para que o texto do projeto tenha uma boa clareza e linearidade. A regra é definida então como:

*7-A ordem dos membros de **LD** é definida pela importância dos elementos para o SOE, devendo começar com aqueles que fazem descrições das Entidades de maior relevância, colocando na seqüência as Entidades relacionadas por ordem de importância.*(BAÑOS, 1997)

A última regra para criação da lista de dependências refere-se às dependências funcionais facultativas, onde o projetista deverá decidir por utilizar uma especialização ou não. Apesar de ser uma regra não obrigatória, segundo (BAÑOS, 1997), *a especialização enfatiza as diferenças entre as entidades pertencentes ao conjunto por meio do estabelecimento de diferenças nos conjuntos de entidades de nível inferior.*(SILBERSCHATZ & KORTH & SUDARSHAN, 1999). Para este caso é definida então a seguinte regra:

8-As descrições das DS Facultativas ficarão mais claras, utilizando-se predicados específicos, por exemplo:

.- quando mais que um.

.- se acaso um etc.(BAÑOS, 1997)

Segundo (BAÑOS, 1997) as regras enumeradas 1, 2, 3, 5 e 6 são invioláveis. As demais são recomendações para simplificar a tradução(T2) de L1 para L2.

2.6.4 - Atividades associadas ao 2º Passo

Neste passo o projetista deverá analisar as frases montadas no passo anterior e criar Diagrama de Dependências (DD) que represente a **LD**. Durante a criação do DD o projetista estará questionando sempre a veracidade das dependências descritas na LD e, conseqüentemente, validando-as. O DD é construído exatamente de acordo com a descrição de S em L1.

A tradução da **LD** se fará segundo as regras de tradução que formam **T2**.

2.6.5 - T2.-Regras para obter um DD a partir da LD.

O diagrama de dependências é um diagrama demonstrativo das dependências funcionais existentes no SOE, representando unicamente todos os atributos que o compõe e suas cardinalidades. A grande preocupação nesta etapa é conseguir representar as regras do negócio em um único diagrama, possibilitando assim uma visão global do sistema para uma melhor análise e conseqüentemente um melhor sistema de informações. Assim como em T1, as regras definidas são simples e fundamentam-se nas dependências funcionais e nos axiomas de Armstrong.

Inicialmente, assim como em T1, o projetista deverá trabalhar de forma ordenada e, partindo do princípio que em T1 já foi realizada uma determinada ordenação, ele deverá respeitar o que já está realizado por T1, para tanto foi definida a seguinte regra:

1-O DD deve ser construído “traduzindo” cada membro da LD, ordenadamente(BAÑOS, 1997)

Também é de responsabilidade do projetista validar as dependências funcionais definidas em T1, assim como sua cardinalidade, evitando representações ilógicas que possam vir a prejudicar a eficiência e integridade do sistema. Caso alguma falha seja encontrada deve-se resolvê-la, atualizando a LD(T1). Para este caso foi definida a regra:

*2-Ao representar cada membro da **LD** será questionado:*

a).- As dependência recíprocas para as DMV. (representado-as em linhas descontínuas se estas forem DS, se são DMV não são representadas);

*b).- As possíveis dependências da Entidade que se está representado (sujeito) com respeito a outras já representadas (Incluindo-as na **LD** se aparecer alguma).(BAÑOS, 1997)*

Vale ainda ressaltar que na regra 2 deste passo, o autor trabalha com os dois tipos de cardinalidade possíveis, ressaltando que em “a” ele solicita ao projetista para que exista o questionamento sobre a cardinalidade da dependência funcional inversa a que está sendo representada, ou seja, caso esteja se representando uma dependência de A para B, o projetista deverá também analisar a dependência de B para A, onde, caso a primeira seja multivalorada a segunda deverá ser também multivalorada, onde, neste caso não será representada. Caso a dependência B para A seja simples o projetista deverá representá-la com uma linha tracejada para posterior análise. Já em “b” existe a preocupação de verificar a sequência de dependências funcionais já levantadas, tanto na LD quanto no DD, questionando sobre novas dependências ainda não levantadas. Caso novas dependências sejam encontradas a LD deverá obrigatoriamente ser atualizada, conforme regra 5, que será vista posteriormente. Esta regra constitui um mecanismo de retro-alimentação do procedimento.

Quando ocorrer a montagem do diagrama a representação de cada dependência funcional deverá estar bem nítida, evitando sobreposição de arcos indicativos ou grandes distâncias entre os atributos que se relacionem. Este ponto está definido de forma simples na seguinte regra:

3-Os atributos com dependências mútuas serão colocados uns próximos aos outros.(BAÑOS, 1997)

Quando da montagem da LD, o projetista deve ter observado a regra 2 de T1, onde o autor solicita que cada frase seja marcada com um número seqüencial. A cada frase da LD representada no diagrama deverá ser anotado sobre os arcos das dependências funcionais o número da frase utilizada. Isso possibilita ao projetista analisar individualmente cada dependência e, em caso de erros encontrados, facilmente atualizá-la na LD. A regra definida para tal necessidade é a seguinte:

*4-Cada dependência representada (DS ou DMV) levará o número de ordem do membro da **LD** que a descreve sobre seu arco correspondente no **DD**. Isto indica as referências cruzadas entre a **LD** e o **DD**.(BAÑOS, 1997)*

Como já exposto quando da explicação da regra 2, o projetista deverá, além de verificar as dependências funcionais já representadas, validar todo o modelo criado, incluindo atributos, cardinalidade e condicionalidade. Caso falhas sejam encontradas o modelo deverá ser atualizado, tanto no diagrama que está sendo criado quanto na LD já definida. Para tanto foi definida a seguinte regra:

*5-À medida que se avança no **DD** devem ser verificados os passos anteriores (Inclusive a **LD**), atualizando a versão, quando necessário.(BAÑOS, 1997)*

Nesta técnica o autor tenta trabalhar da forma mais simples possível, facilitando assim a compreensão da técnica, assim como a sua aplicação. Com respeito ao diagrama não é diferente, onde a representação mais simples deverá ser utilizada sempre que possível evitando assim o uso desnecessário de representações rebuscadas, o que é o caso de agregações desnecessárias. Esta normatização é indicada através da seguinte regra:

*6-Ao considerar as alternativas de representação de cada membro da **LD**, eleja sempre a que produza o **DD** mais simples. (BAÑOS, 1997)*

Assim como não podem ser utilizados sinônimos na LD, no diagrama de dependências cada atributo somente poderá ser representado uma única vez. Isso garante ao projetista um nível nulo de redundâncias em dicionário, onde não haverão repetições de atributos, podendo, portanto ser utilizado em ferramentas CASE's. A regra a seguir define este caso de identificação de atributos.

7-*Cada identificador de atributo (ou grupo) representante de uma Entidade ou de uma propriedade de Entidade, somente pode aparecer uma vez no DD quando tenha o mesmo domínio. Caso o domínio seja diferente podem ser criados outros identificadores. Cuidado com o uso de sinônimos. (BAÑOS, 1997)*

Uma das ocorrências mais críticas em processamento de dados é o chamado “estado de deadlock”, que ocorre, segundo SILBERSCHATZ & KORTH & SUDARSHAN, (1999) *quando há um conjunto de transações, tal que toda a transação desse conjunto está esperando outra transação também nele contida.* Uma forma de se prevenir o estado de deadlock é evitando ciclos entre dependências funcionais de atributos, pois uma vez que não há ciclos, o sistema não estará em estado de deadlock (SILBERSCHATZ & KORTH & SUDARSHAN, [1999 p495]. Nesta técnica a prevenção do deadlock é tratada através de uma regra que minimizará seus efeitos. Vale ressaltar que o ciclo estará formado desde que exista a ligação através de dependências funcionais diretas entre atributos de forma fechada, mesmo estando os arcos direcionais apontando para direções opostas, pois as dependências funcionais sempre serão mútuas, ou seja, se A determinar B, B irá de alguma forma levar a A, mesmo que seja através de dependência Multivalorada, o que é definido na regra 9. Quando diz-se que a adoção desta regra minimizará e não resolverá o problema é porque ainda poderão existir “deadlocks”³ originários de processos mal projetados, o que não é o foco desta técnica. Portanto, a seguinte regra é definida:

8.-*Não podem ser criados “ciclos” no DD. No caso de aparecerem devem ser quebrados. (BAÑOS, 1997)*

Exemplo:

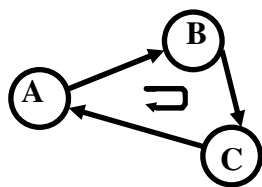


Figura 13 – Representação de Ciclo (Deadlock)

³ Deadlock – Termo utilizado para indicar um laço de busca infinito em um Banco de Dados, em virtude de bloqueios simultâneos de registros necessários para uma determinada tarefa, que leva ao consumo excessivo de hardware comprometendo, na maioria das ocorrências, a performance do software, obrigando uma inicialização de todo o sistema para quebrar o processo.

Como exposto inicialmente na definição da regra anterior, as dependências entre atributos sempre serão mútuas, e deverão ser representadas unicamente, escolhendo sempre a dependência simples sobre a dependência multivalorada, ou melhor, obedecendo a regra 6 que solicita que seja utilizada sempre que possível a representação mais simples, pois é claro que uma dependência multivalorada possui como retorno outra dependência multivalorada e jamais uma dependência simples, pois, caso isso ocorra a dependência simples deverá ser demonstrada para melhorar a garantia de integridade e navegabilidade do banco de dados. Neste caso específico, a dependência multivalorada poderá ser definida, em nível físico, através da criação de uma visão de banco de dados, o que não comprometerá a integridade do sistema. Então:

9-As dependências entre qualquer par de atributos ou grupo deles é sempre mútua, porém somente se representa uma delas no DD. (BAÑOS, 1997).

Para uma representação mais simples do diagrama, visto que muitos atributos serão representados, existe uma preocupação na ordenação destes atributos, em complemento à regra 3, onde não é suficiente colocar os atributos com dependência mútua próximos, mas também ordená-los, conforme a seguinte regra:

10.-As dependências representadas criam “cadeias” entre os diferentes círculos. O sentido destas cadeias tem que ser aquele que:

a).- Inicie dos atributos de maior importância relativa.

b).- Produza mais dependência simples na cadeia.

c).- Produza menos valores NULL nas cadeias. (BAÑOS, 1997)

Conforme a regra 7, cada atributo poderá ser representado apenas uma vez no DD e, conseqüentemente, várias dependências funcionais poderão chegar até ele, principalmente se este atributo fizer parte de entidades de relacionamento. Neste caso, para efeito de melhor visualização e posterior mapeamento, devem ser aplicados múltiplos círculos na representação, um para cada dependência funcional representada. Esta regra fala que para toda dependência funcional deverá ser criado um novo círculo, porém, através da experiência de uso notou-se que somente as dependências funcionais multivaloradas é que é

necessário serem representadas com múltiplos círculos para posterior análise de criação de chaves primárias, o que será analisado nas regras do passo 3(T3). A regra 11 define este estado.

11.-*Não podem chegar dois ou mais arcos distintos a um mesmo atributo ou grupo, a não ser que sejam utilizados múltiplos círculos em sua representação.* (BAÑOS, 1997)

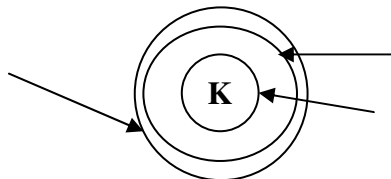


Figura 14 – Representação de alvos com recepção múltipla de arcos

Em virtude da proibição da repetição do uso de atributos, imposto pela regra 7, o projetista poderá encontrar dificuldade em representar atributos que possuam o mesmo domínio em diferentes cadeias, o que deverá ser resolvido marcando-os com a chamada “bandeira de domínio”, representada por um triângulo com o número do domínio que está sendo utilizado ao centro.

12.-*Quando dois ou mais atributos ou grupos têm domínio comuns, o assinalamos no DD colocando uma bandeira de domínio para cada domínio em comum..*(BAÑOS, 1997)



Figura 15 – Representação de Bandeira de Domínio

As regras numeradas de 7 a 12 são invioláveis.(BAÑOS, 1997)

2.6.6 - Atividades do 3º Passo.

Após a etapa de levantamento de dados junto ao usuário, o desenvolvimento do projeto através de diagramação e validação das informações, o projetista deverá então criar efetivamente suas tabelas básicas para a montagem física da base de dados. Nesta etapa o DD deverá ser traduzido com a finalidade de serem identificadas todas as tabelas do sistema e seus relacionamentos. Vale ressaltar que nesta base do modelo físico são representadas inclusive as tabelas de relacionamento, originárias de relacionamentos expressos com cardinalidade N:N em diagrama entidade relacionamento (DER), com seus respectivos atributos. A representação das tabelas se dará seguindo as regras de T3, que têm como objetivo obter BD' para o SOE como expressão de S em L3. (BAÑOS, 1997)

2.6.7 - T3.-Regras para obter as Tabelas Básicas.

A terminologia aplicada a T3 coloca como alvo todo atributo que receber um arco simples, ou melhor, todo atributo que for dependente funcionalmente de outro através de uma dependência funcional simples. Estes serão identificados através da letra A, originária da palavra Alvo, com um número seqüencial relacionado à sua origem, ou seja, todos os atributos dependentes funcionalmente da mesma origem receberão a mesma identificação, definida pela seguinte regra:

1- Cada círculo alvo será identificado com um A seguido de um número consecutivo como segue:(BAÑOS, 1997)

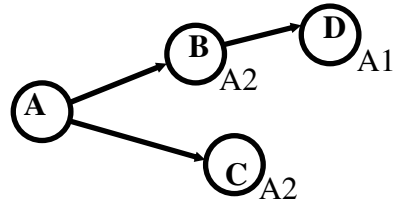


Figura 16 – Representação de Diagrama de Dependências.

No exemplo acima existem 3 círculos alvos porém somente 2 diferentes, pois tanto B como C são propriedades de A. Caso C seja Facultativa de A será identificado com F1 e não A2, conforme será visto posteriormente.

Outra definição desta técnica são as “end-keys”, ou chaves terminais, que originarão as tabelas de relacionamentos que possuem somente as chaves como propriedade, ou seja, não possuem atributos. As “End-keys” são atributos que são dependentes funcionalmente de outro atributo com cardinalidade múltipla e não possuem nenhum atributo dependente funcionalmente delas. Estes círculos serão marcados com a letra EK, seguida por um número de ordem seqüencial, independente do atributo de origem.

2- Cada círculo de chave terminal será identificado com um EK (End-Key) também seguida de um número consecutivo. Uma EK é um Atr com dependência multivalorada que não indica nenhum alvo. (BAÑOS, 1997)

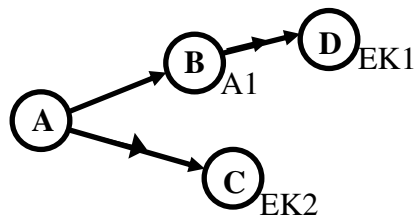


Figura 17 – Representação de Diagrama de Dependências com chave terminal.

Assim como em T1 e T2, uma organização das tarefas do projetista é importante, pois não é aconselhável trabalhar desordenadamente. Neste caso, a próxima regra determina

a ordem que devem ser traduzidas as dependências funcionais para transformá-las em tabelas, a saber:

3-A *“tradução” do DD será feita resolvendo os círculos na seguinte ordem:*

a).- Alvos (An).

b).- Chaves terminais(EKn).

c).- Círculos isolados.(BAÑOS, 1997)

Cada cadeia de atributos deverá sempre gerar uma única tabela, respeitando a chave primária originária, ou melhor, o atributo que origina todas as dependências funcionais do enlace será identificado como chave primária da tabela. No caso de alvos com origem de atributos com dependência funcional multivalorada, todos os atributos multivalorados comporão a chave primária. As end-keys formarão tabelas sem atributos, somente com a chave primária composta, o que possibilitará ao projetista definir relacionamentos com cardinalidade N:N.

4-Cada cadeia de enlaces gera uma tabela.(BAÑOS, 1997)

Como validação, o projetista não poderá criar tabelas que possuam a mesma chave primária, salvo o caso de atributos facultativos, que darão origem a tabelas com as mesmas chaves primárias que comporão uma especialização de tabelas.

5-Não podem existir duas tabelas de igual chave exceto no caso de Atr Facultativos.(BAÑOS, 1997)

Para uma melhor visualização, o projetista deverá marcar suas chaves primárias e estrangeiras, sendo sugerido que as chaves primárias sejam sublinhadas com linhas contínuas e as chaves estrangeiras com linhas pontilhadas, conforme a seguinte regra:

6-Os atributos que são chaves estrangeiras ou partes delas, serão sublinhados com linhas descontínuas nas tabelas, e os que formam a chave primária, com linha contínua.(BAÑOS, 1997)

Finalmente, para efeito de organização, o projetista poderá marcar no DD todas as cadeias já traduzidas para T3. Como sugestão definiu-se a seguinte regra:

7- Cada cadeia representada como tabela é marca no DD da seguinte maneira:(BAÑOS, 1997)

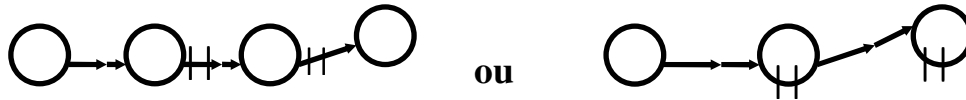


Figura 18 – Representação da marcação de leitura no diagrama de dependências.

2.6.8 - Regras para posicionamento relativo das tabelas básicas da BD.

Após a tradução do DD em tabelas básicas, o projetista deverá ordená-las e relacioná-las para obter uma visão final do seu projeto e possibilitar a criação dos esquemas para a base de dados desejada. Para tanto algumas regras foram definidas por (BAÑOS, 1997), a saber:

1. Eleger a tabela de chave primária mais extensa para colocá-la ao centro do esquema;
2. Buscar, entre as tabelas restantes, a tabela básica com mais atributos comuns (na chave ou fora dela) com a tabela eleita anteriormente, colocando-as junto;
3. Agora se tem dois “extremos”. Repetir o 2º Passo comparando cada vez com cada extremo até ter posicionado todas as tabelas básicas;
4. Unir, através de traços, os atributos de igual nome (ou domínio) caso se pretenda navegar por eles entre as tabelas básicas para determinar consultas.

5. Sublinhar as Chaves primárias (PK) com um traço contínuo e as chaves estrangeiras (FK) com um tracejado, conforme regra 6 de T3.

Estas regras são usadas para obter um documento de registro da distribuição das tabelas básicas que permita refletir com clareza os relacionamentos de navegação entre elas. Abaixo segue um exemplo da aplicação de T3.

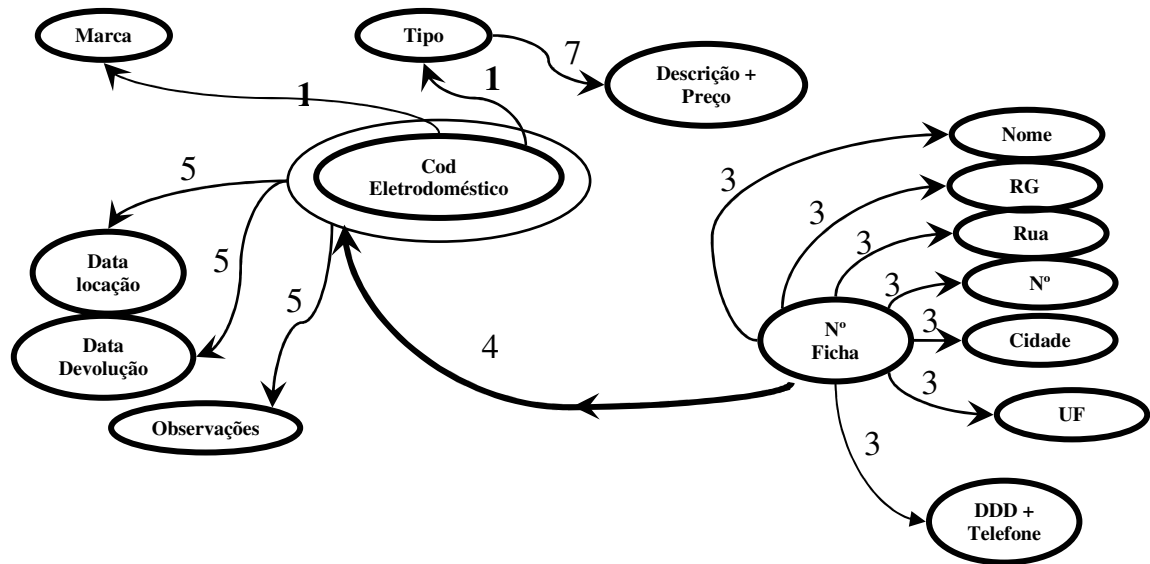


Figura 19 - Exemplo de Diagrama de Dependências

Uma vez expressado **S** em **L2 (DD)**, podem-se aplicar as regras que estabelecem **T3** e sua ordenação, quando serão obtidas as seguintes tabelas:

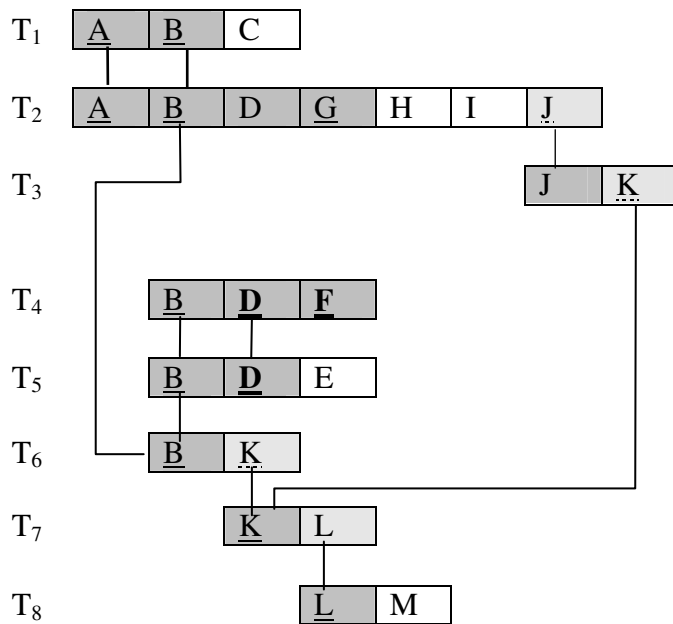


Figura 20 - Modelo de Tabelas Básicas Interligadas

Os atributos sombreados com a tonalidade mais forte são componentes de Chaves Primárias, os que aparecem mais claros representam Chaves estrangeiras ou Bandeiras de Domínio. As bandeiras de domínio são colocadas nos atributos que possuem o mesmo domínio. Deve-se levar em conta nesta etapa de projeto as Chaves Estrangeiras e as bandeiras de domínio, pois permitem estudar a ligação entre as tabelas por Chaves Estrangeiras e as possíveis ligações entre tabelas por Bandeira de Domínio. As bandeiras de domínio são simbolizadas no DD com um triângulo ao que se coloca em seu interior um número que indica a ordem que tem a ligação entre todos os atributos desse domínio.

As linhas que unem as tabelas são denominadas ligações de navegação. Elas garantirão na etapa de desenvolvimento as ASSOCIAÇÕES entre as tabelas.

2.7 – *Inteligência Artificial*

Criar máquinas semelhantes ao homem com capacidade de pensar sempre foi um fascínio que vem atraindo cientistas do mundo obstinados à realização deste sonho. Os mais antigos escritores já esboçavam robôs e andróides em suas obras de ficção e hoje cada vez mais são produzidas obras que ilustram bem a ansiedade do homem criar máquinas capazes de auxiliá-lo nas mais difíceis tarefas do dia-a-dia. Tarefas básicas e repetitivas já são realizadas, em grandes indústrias, por robôs. Essas tarefas produzem sempre resultados precisos e iguais, não tendo portando a máquina que resolver novos problemas. A máquina que pensa parte basicamente da premissa que o homem consegue resolver o problema de forma eficiente. O principal dilema está em que o homem consegue resolver o mesmo problema de N formas diferentes. Segundo RICH & KNIGHT (1991), “*A Inteligência Artificial é o estudo de como fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor*”.

Etimologicamente a palavra *inteligência* vem do latim inter(entre) e legere(escolher), ou seja, inteligência significa aquilo que nos permite escolher entre uma coisa e outra. A palavra *artificial* também deriva do latim artificiale, que significa algo não natural, isto é, produzido pelo homem. Portanto, inteligência artificial é um tipo de inteligência produzida pelo homem para dotar as máquinas de algum tipo de habilidade que simula a inteligência do homem. (TORRE, 2001).

A Inteligência Artificial busca ainda entender a mente humana e imitar seu comportamento levantando questões como: “Como ocorre o pensar?”, “Como o homem extrai conhecimento do mundo?”, “Como a memória, os sentidos e a linguagem ajudam o desenvolvimento da inteligência?”, “Como surgem as idéias?”, “Como a mente processa informações e tira conclusões decidindo por uma coisa ao invés da outra?”. Essas são algumas perguntas que a inteligência artificial precisa responder, enquanto ciência, para simular o raciocínio humano e implementar aspectos de inteligência nas máquinas.

A Inteligência Artificial pode ainda ser definida como um conjunto de técnicas e metodologias de programação usadas para tentar resolver os problemas de forma mais eficiente que soluções algorítmicas, e fazendo isso o mais próximo possível de um ser humano, ou seja, o primordial objetivo da Inteligência Artificial é tentar fazer com que um computador consiga processar informações da mesma forma que um ser humano as processa.

Pode-se definir o tipo de processamento de uma abordagem do estilo da Inteligência Artificial de seguinte forma:

- a) Recupera uma instrução da memória;
- b) Recupera também alguns dados que sejam necessários ao processamento da instrução;
- c) Executa a instrução;
- d) Guarda os resultados em memória;
- e) Retorna ao passo “a”.

Como aplicação, alguns especialistas acreditam que nas próximas décadas a pesquisa em Inteligência Artificial será capaz de criar robôs computadorizados que poderão superar os humanos em todas as capacidades intelectuais, porém, atualmente, sistemas inteligentes gerenciam a qualidade da produção em refinarias de petróleo, controlam a geração e distribuição de energia elétrica em usinas, organiza ambientes hospitalares, dentre outras atividades que variam de controles de áreas industriais a gestão de recursos ambientais. Como campos de aplicação, pode-se citar:

- a) Processamento de Linguagem Natural – Voltado para a construção de programas capazes de compreender a linguagem natural humana (interpretação) e gerar textos.
- b) Reconhecimento de Padrões – É uma das áreas de pesquisa mais avançadas em Inteligência Artificial, que estuda a capacidade de reconhecimento de padrões através de programas que tratem voz, dígitos, escrita, imagens, etc.

- c) Visão de Computador – Busca desenvolver formas do computador trabalhar com visão bidimensional e tridimensional.
- d) Programação de Jogos – Voltado para a construção de programas de entretenimento que envolva raciocínio lógico. Os jogos computadorizados exibem um tipo de inteligência capaz de desafiar as habilidades do jogador. O jogo de Xadrez, por exemplo, foi utilizado para as primeiras experiências em programação do raciocínio artificial, onde o computador se tornou capaz de analisar milhões de jogadas por segundo para tentar derrotar seu adversário. Além de analisar as jogadas, os programas utilizam um método heurístico que consiste na utilização de uma árvore de busca que possui ramificações a partir de certos nós. Estes nós representam os pontos de decisão do caminho que deve ser tomado para se alcançar um objetivo.
- e) Robótica – É o campo de estudo voltado a desenvolver meios de construir máquinas que possam interagir com o meio (ver, ouvir e reagir a estímulos sensoriais). A expressão robô vem do tcheco “robota”, que significa trabalhador e foi criada por Karel Capek, em 1917.
- f) Aprendizado – Existem programas de Inteligência Artificial que conseguem aprender certos fatos por meio de experiência, desde que esse conhecimento possa ser representado de acordo com o formalismo adotado pelo programa.

Para efeito desta pesquisa, a área mais pertinente ao processo de análise de banco de dados diz respeito à análise de linguagem natural, não desprezando porém outras áreas como a de aprendizado.

2.7.1 - Processamento da Linguagem Natural

O processamento de linguagem natural está relacionado ao desenvolvimento de modelos computacionais nos aspectos do processamento da linguagem humana, como, por

exemplo, a leitura e interpretação de um texto, escrita de uma carta, um conversa, tradução de um documento, procura por informações importantes, etc. Estes modelos são úteis para desenvolvimento de programas que executam importantes tarefas envolvendo o processamento de linguagem para obtenção de uma melhor compreensão da comunicação humana. Corretores de ortografia e gramática, sistemas de busca de informação, sistemas de reconhecimento de voz e interfaces de linguagem natural são exemplos de tais programas.

O Processamento de Linguagem Natural, ou simplesmente PLN, é composto por idéias de várias áreas do conhecimento, como Lingüística, Filosofia, Psicologia, Inteligência Artificial e Ciência da Computação. Os lingüistas necessitam entender a própria linguagem, psicólogos querem entender o processo de compreensão e produção, filósofos querem descobrir como pensamentos racionais se relacionam com a linguagem e como as palavras se tornam significativas, estudiosos de inteligência artificial procuram desenvolver modelos de raciocínio humano envolvendo processamento de linguagem, e finalmente, cientistas da computação querem desenvolver aplicações melhores que envolvam o processamento de linguagem natural e melhorem a interface homem-máquina.

Atualmente existem duas principais aplicações de Processamento de Linguagem Natural: o processamento de texto e a máquina de comunicação humana. O processamento de texto escrito tem como objetivo oferecer ao autor humano a checagem de gramática e estilo. Pode ainda extrair informações de reportagens automaticamente, produzindo um sumário. Classifica, retorna ou filtra mensagens ou documentos, traduz documentos ou gera relatórios textuais a partir de informações de um banco de dados. Como exemplo da máquina de comunicação humana podem-se citar as interfaces de linguagem natural para pesquisa em banco de dados, serviço ao consumidor automatizado através do telefone, máquinas controlados por comando de voz e sistemas tutoriais. Tais sistemas devem lidar com linguagem informal diária, e manter um diálogo com o usuário.

O Processamento de Linguagem Natural, neste trabalho, será considerado apenas como uma sub-área da Inteligência Artificial envolvida na Ciência da Computação que têm como foco o desenvolvimento de algoritmos inteligentes, estruturas de dados e as mais diversas formas de representação de conhecimento que sejam utilizadas no tratamento de linguagem nos sistemas computacionais através das suas duas vertentes principais: Geração de Linguagem e Compreensão de Linguagem.

Desde a sua criação, os computadores têm auxiliado o homem a executar as suas tarefas rotineiras, principalmente as que podem ser traduzidas em uma linguagem programação do computador, e este então faz o que lhe foi solicitado. O atual e maior problema encontrado nesta situação está no uso de linguagens de programação que seguem um formalismo necessário para que o computador possa interpretá-la e processá-la. Muitos são os estudos de tratamentos que possam fazer com que o computador entenda linearmente a linguagem natural, escrita ou falada, porém, segundo RICH & KNIGHT (1991) existem vários problemas lingüísticos que dificultam esta atividade, como:

- As diversas formas de se expressar o mesmo fato;
- As diversas interpretações de uma mesma sentença, aplicadas a um contexto;
- O uso de sentenças incompletas que contam com o conhecimento do interlocutor;
- A agregação de novas palavras ao cotidiano do ser humano.

Uma das áreas de grande interesse da Inteligência Artificial está voltada para o uso da Linguagem Natural na programação de instruções no computador. Esta linguagem possibilitaria ao ser humano escrever, ou simplesmente ditar uma instrução, por exemplo, sem se preocupar com o formalismo da linguagem de programação, o que já é possível através do uso de programas específicos para reconhecimento de linguagem natural, como o “Via Voice” da IBM, porém todas as instruções somente podem ser dirigidas a um ambiente controlado de gerenciamento de atividades do sistema, estando ainda longe do desejado sistema operacional amigável, controlado através de comandos em linguagem natural.

Em uma primeira análise, este processamento não aparenta muitas dificuldades para o seu desenvolvimento, visto que utilizamos este tipo de linguagem desde o início de nossa vida, porém existe uma complexidade muito grande com relação a este tipo de processamento, relativa à língua, contexto, gírias e costumes, etc.

Para se utilizar a linguagem natural em uma análise RICH & KNIGHT (1991) colocam que o processamento da linguagem natural pode ser dividido nas seguintes etapas:

- Análise Fonética

- Análise Morfológica
- Análise Sintática
- Análise Semântica
- Integração do Discurso
- Análise Pragmática

A etapa de análise fonética reconhece sílabas a partir dos sons que formam um enunciado.

Na etapa da análise morfológica verifica-se a existência da palavra e as suas variações. Esta análise é feita isolando-se a palavra dos componentes não textuais, como, por exemplo, pontuação.

Na etapa de análise sintática as palavras são convertidas em estruturas, onde se verifica a sua relação com respeito às demais. Isto somente é obtido depois de definidas as regras de gramática da linguagem utilizada, onde um agente terá a função de validar o emprego gramatical correto das palavras nas instruções propostas.

Na etapa de análise semântica é produzida uma estrutura de análise geralmente em formato de árvore cujas folhas são palavras e os nós internos, as categorias. A partir das folhas informações são programadas até a raiz para produzir uma representação do conteúdo vinculado pela frase. Cada nó desta árvore representa uma parte da frase, que pode ser o sujeito, o verbo de ligação ou o predicado verbal.

Na etapa de integração do discurso, a frase é interpretada juntamente com outras frases do texto para que seja detectado sujeito não explícito, como é o caso de pronomes.

E finalmente, na etapa de análise pragmática é a etapa onde, após adquirir o conhecimento do texto, o sistema poderá traduzir, quando necessário, os comandos determinados neste conhecimento em linguagem de máquina capaz de instruir o computador. (RICH & KNIGHT, 1991)

3. - Proposta de Solução

3.1 - Definição das Regras

Conforme descrito anteriormente, a técnica de lista diagrama está baseada em regras rígidas que forçam o projetista ao total conhecimento lógico do sistema que se deseja desenvolver, principalmente as informações, dados e dependências existentes neste.

Para criação do protótipo da análise textual, as regras devem ser definidas e testadas. Na criação do sistema de avaliação serão utilizadas como base as regras para criação da lista de dependências definidas no modelo Lista-Diagrama, porém, como estas regras necessitam da complementação de análise, o completo entendimento do sistema será alcançado somente com a análise do diagrama de dependências criado a partir desta lista, o que não é objeto deste trabalho, podendo ser desenvolvido posteriormente em evolução a esta dissertação.

3.2 - Definição dos Processos

Para a criação da ferramenta de análise serão consideradas as regras do passo 1, criação de lista de dependências, dentro do modelo lista-diagrama, citadas anteriormente e, para cada regra será realizada uma análise de quais etapas da PLN deverão ser utilizadas. Inicialmente cada frase deverá estar numerada seqüencialmente, atendendo a Regra 2 do modelo. Isso será alcançado através da Interface do usuário.

Para o desenvolvimento da análise de textos no projeto serão consideradas as seguintes etapas:

- Entrada de Dados
- Análise Morfológica
- Análise Sintática
- Análise Semântica
- Integração do Discurso
- Análise Pragmática
- Impressão de relatório final de análise

3.3 - Entrada de Dados

Inicialmente o projetista deverá fazer um breve cadastro do sistema que está sendo proposto para efeito de documentação. Estes dados não terão efeito na análise automatizada, porém serão utilizados no momento da geração dos relatórios técnicos relativos ao sistema.

Os dados solicitados são:

- Nome – Identificação do Sistema que será analisado.
- Descrição – Uma breve descrição dos objetivos do sistema. Neste campo o autor não deve se preocupar com detalhamento do sistema, mas sim com um texto sucinto de introdução ao sistema.
- Autor – Nome do autor do Sistema.
- Versão – Versão da Documentação. Deverá ser atualizada automaticamente a cada validação do sistema. O valor inicial será 1.

- Data da Versão – Data da última análise do sistema onde foi alterada a versão. A atualização deste campo será automática, vinculada à alteração da versão.

Lista de Dependências
Cadastro do Sistema

Nome

Descrição

Autor

Versão

Data da Versão

Figura 21 - Protótipo da Tela de Cadastramento de Sistema

Após a identificação do sistema o projetista deverá indicar quais são as regras de negócio que o regem, respeitando as normas descritas no passo 1 da criação de lista de dependências. Para esta tela, é proposto um protótipo dividido em duas partes. Na primeira o projetista cadastrará as regras pertinentes, separadas por frases e na segunda parte serão visualizadas as regras já cadastradas no sistema.

Após o cadastramento das regras, o sistema armazenará cada uma no estado original, possibilitando ao analista uma futura comparação do estado original e o estado após análise. Isso é necessário por dois motivos:

- O analista poderá corrigir suas falhas e trabalhar com maior precisão em próximos levantamentos;

- Permitir uma avaliação de confiabilidade da ferramenta, visto que podem ser obtidos resultados inconsistentes, pois podem existir falhas nas regras do sistema;

Os campos necessários para o armazenamento serão:

Sistema – Campo informativo que indica ao projetista o nome do sistema onde ele está trabalhando. Campo não atualizável.

Versão – Informa a versão atual do projeto. Atualizado automaticamente a cada pedido de análise. Campo não atualizável.

Regra – Espaço onde o analista escreverá suas regras de negócio, respeitando a exigência de formato solicitada pela norma quatro da técnica de lista-diagrama, ou seja, na frase deverá estar bem definido o sujeito, o verbo e o seu predicado. Para garantir esta precisão, a frase foi dividida em partes, a saber:

- a) Identificador normal ou facultativo do sujeito da frase.
- b) Sujeito da Frase
- c) Verbo que indica o(s) atributo(s) de Identificação do Sujeito.(opcional)
- d) Atributo identificador do sujeito. (opcional)
- e) Verbo de ligação da Frase
- f) Atributos do sujeito.

As especializações deverão ser tratadas com um indicativo de facultatividade no campo “a”, como, por exemplo, “*Caso*”. Isso indicará ao agente uma continuação de frase, onde deverá sempre ser observado qual é o sujeito ao qual esta está fazendo referência.

Regras – Indicam ordenadamente quais regras já estão cadastradas, permitindo a seleção para alteração. Ao ser cadastrada cada uma das regras, o sistema automaticamente a numerará sequencialmente. Este número será utilizado posteriormente na análise de importância das frases.

Lista de Dependências
Regras do Negócio

Sistema Versão

Regra (a) (b) (c) (d)

(e) (f)

Salvar **Anterior** **Próximo**

Regras

01 – Cada cliente é identificado através de um código numérico seqüencial e tem como atributos seu nome, e telefone. Caso o cliente seja Pessoa Física ele terá ainda os atributos CPF, data de nascimento e sexo. Caso o cliente seja Pessoa Jurídica ele terá como atributos o CNPJ, nome de fantasia e Inscrição Estadual.

02 – Cada cliente empresta vários Livros.

03 – Todo livro é identificado pelo ISBN, tem um título e uma editora.

Alterar **Analisar** **Reordenar** **Imprimir**

Figura 22 - Protótipo da Tela de Cadastramento das Regras do Negócio

O botão Salvar permitirá ao usuário indicar ao sistema que sua frase está completa e poderá ser armazenada no sistema. O processo de armazenamento deverá então numerar a frase segundo o número de seqüência desta, respeitando:

a) A seqüência numérica crescente das frases informadas anteriormente neste sistema. Caso não existam frases anteriores, o número “01” será indicado.

b) Não numerar as frases que iniciem com indicadores de facultatividade. Neste caso a frase que está sendo salva deverá ser acrescida a uma frase informada anteriormente onde o sujeito seja idêntico em ambas e nenhuma indique dependências multivaloradas. Caso não exista uma frase anterior o sistema deverá indicar um erro solicitando que seja informada uma frase não facultativa e, posteriormente, a opcional ou solicitar que seja alterado o indicador de facultatividade da frase.

Os botões *Próximo* e *Anterior* permitirão a escolha seqüencial das frases já informadas, permitindo a edição.

Na parte inferior da tela existem ainda quatro botões que são o de Análise, o de alteração, o de reordenação e o de Impressão.

Na opção de Alterar o sistema permitirá ao usuário editar a frase escolhida alterando-a conforme sua necessidade.

Na opção de Analisar, o sistema verificará cada uma das frases propostas, realizando as tarefas de análise textual que serão descritas posteriormente.

Na opção Reordenar o sistema permitirá ao usuário indicar uma nova ordem de importância para as frases já cadastradas.

<i>Lista de Dependências</i>	
<i>Ordenação de Importância</i>	
Sistema <input type="text"/>	Versão <input type="text"/>
01	Cada cliente é identificado através de um código numérico seqüencial e tem como atributos seu nome, e telefone. Caso o cliente seja Pessoa Física ele terá ainda os atributos
02	Cada cliente empresta vários Livros.
03	Todo livro é identificado pelo ISBN, tem um título e uma editora.
04	Toda editora é identificada pelo seu CNPJ e tem como atributos nome, telefone e pessoa de contato.
<div>Imprimir</div> <div>Sair</div>	

Figura 23 - Protótipo da Tela de Reordenação das Regras do Sistema

Na opção imprimir, o usuário poderá imprimir as regras do sistema, no seguinte formato:

<i>List a de Dependências</i>	
<i>Regras Do negócio</i>	
Sistema:	<i>Sistema de Empréstimo de Livros</i>
Descrição:	<i>Este sistema terá como objetivo atender às necessidades básicas de empréstimos de livros dentro de uma Biblioteca.</i>
Autor:	<i>Paulo Ricardo T. Diniz</i>

Análise dos Dados

Figura 24 - Protótipo do Relatório de Regras cadastradas no sistema

Cada uma das opções do sistema estará integrado através de agentes de validação que serão baseados nas 8 regras da etapa de lista de dependências.

Para a programação será utilizada a linguagem Java por permitir o uso nativo das características de Orientação a Objetos, principalmente hierarquia e herança, pois os agentes deverão trocar entre si as informações das regras que estão sendo projetadas e estas características possibilitam o controle do sistema e o ajuste deste conforme as regras forem sendo incluídas no sistema. Outra justificativa aplicável ao uso do Java está na sua portabilidade, pois existe a intenção de que este sistema seja disponibilizado para uso via Web e, neste caso todos os agentes serão reaproveitados sendo necessário apenas o desenvolvimento de uma nova interface.

3.4 - Cadastramento das Regras

No momento do cadastramento das regras alguns cuidados deverão ser observados para que seja alcançado um bom índice de confiabilidade do sistema. Em um primeiro

estudo pensou-se em aceitar a entrada de frases puras, ou seja, da forma que os usuários as cadastrassem, porém no momento do projeto de protótipo muitas foram às formas encontradas para que esta entrada ocorresse e, conseqüentemente, muitas seriam as formas de análise da frase o que inviabilizaria o desenvolvimento de um bom nível de confiabilidade para o sistema, além de fugir do objetivo básico deste projeto que é o de ensinar a escrever boas descrições de modelos conceituais de Banco de Dados. Por estes motivos optou-se por trabalhar em um ambiente mais controlado, ou seja, efetuando alguns dos controles necessários na entrada dos dados. Para isso foi definida uma estrutura padrão de entrada de dados que permita a definição clara dos sujeitos (entidades) e seus atributos.

Como exemplo de uma definição de regras não claras, pode-se citar uma especificação encontrada em SILBERSCHATZ & KORTH & SUDARSHAN, 1999 pags. 48,49 e 50, onde os autores propõem o seguinte enunciado para análise:

“A descrição resultante dessa fase do projeto é a base para a especificação da estrutura conceitual do banco de dados. A lista de itens que se segue apresenta as principais necessidades de uma empresa da área bancária”.

- *Um banco é organizado em agências. Cada agência é localizada em uma cidade e é identificada por um nome único. O banco controla os fundos de cada uma dessas agências.*
- *Os clientes do banco são identificados pelo número de seu seguro social. O banco mantém dados como nome, rua e cidade do cliente. Os clientes podem possuir contas e contrair empréstimos. O cliente pode estar associado a um bancário específico que cuida de seus negócios ou atua como um agente de empréstimos.*
- *Os empregados do banco também são identificados por meio do número de seu seguro social. A administração do banco mantém o nome e o número do telefone de cada um de seus empregados, os nomes de seus dependentes e o número do seguro social de seu gerente. O banco também possui a data de contratação do empregado e, com isso, seu tempo de trabalho.*
- *O banco oferece dois tipos de contas - contas poupança e contas movimento. As contas movimento podem possuir mais de um correntista, e um correntista pode possuir mais de uma conta. Cada conta possui um único número. O banco controla o saldo de cada conta, assim como a data mais recente de acesso a essa conta. Por outro lado, cada conta poupança possui a taxa de juros associada, assim como são também registrados os excessos nos limites da conta movimento.*

- *Um empréstimo originado em uma agência em particular pode ter sido obtido por um ou mais clientes. Um empréstimo é identificado por um número único. Para cada empréstimo o banco mantém o montante emprestado, assim como os pagamentos das parcelas. Embora o número da parcela de um empréstimo não identifique de modo único um pagamento específico dentre os muitos realizados no banco, o número da parcela identifica um pagamento efetuado para um empréstimo em particular. A data e o montante são registrados no pagamento de cada parcela.*

Em um banco real, poderia ser de interesse manter informações sobre depósitos e retiradas tanto para as contas poupança quanto para as contas movimento, assim como se mantêm informações sobre o pagamento de parcelas dos empréstimos. Uma vez que a modelagem dos requisitos dos usuários nas necessidades descritas a pouco são semelhantes àquelas feitas no início, podemos optar por um exemplo de aplicação mais reduzido, não fazendo, em nosso modelo, o acompanhamento dos depósitos e das retiradas.”

Como complemento o autor especifica as seguintes entidades.

“... começamos por identificar os conjuntos de dados e seus atributos.

- *O conjunto de entidades agência, com os atributos:*
 - *nome_agência, cidade_agência e fundos.*
- *O conjunto de entidades cliente, com os atributos:*
 - *nome_cliente, seguro_social rua_cliente e cidade_cliente*
Com a possibilidade do atributo adicional nome_bancário.
- *O conjunto de entidades empregado, com os atributos:*
 - *seguro_social_empregado, nome_empregado, número_telefone, salário e gerente .*
Recursos adicionais descritivos são os atributos multivalorados nome_dependentes, o atributo básico data_início e o atributo descritivo tempo_de_trabalho.
- *Dois conjuntos de entidades contas – conta_poupança e conta_movimento - com os atributos comuns número_conta e saldo; também, conta_poupança possui o atributo taxa_de_juros e a conta_movimento, o atributo limite_cheque_especial.*
- *O conjunto de entidade empréstimo, com os atributos:*
 - *número_empréstimo, total e agência_origem*

São possíveis os seguintes atributos adicionais: pagamento_empréstimo, composto multivalorado; com os seguintes atributos componentes: número_pagamento, data_pagamento e total_pagamento “.

Até este estágio os autores definiram uma descrição das entidades e atributos do sistema e, finalmente, definem os relacionamentos, da seguinte forma:

“Retornemos agora ao esquema do projeto rudimentar da Seção 2.8.3 para especificar os seguintes conjuntos de relacionamentos e mapeamentos de cardinalidades:

- *devedor*, como conjunto de relacionamentos muitos para muitos entre cliente e empréstimo.
- *agência_empréstimo*, conjunto de relacionamentos muitos para um que indica qual a agência responsável pelo empréstimo.
- *pagamento_empréstimo*, relacionamento um para muitos entre empréstimo e pagamento, que documenta que um pagamento está sendo feito para um determinado empréstimo.
- *depositante*, com os atributos de relacionamento *data_acesso*, um conjunto de relacionamentos muitos para muitos entre cliente e conta, indicando que um cliente possui uma conta.
- *agente_cliente*, como atributo de relacionamento tipo, um conjunto de relacionamentos muitos para um expressando que um cliente pode ser atendido por determinado empregado do banco e que um empregado do banco pode atender a um ou mais clientes.
- *trabalha para*, conjunto de relacionamentos entre entidades empregado que determina se se trata de gerente ou empregado; o mapeamento da cardinalidade expressa que um empregado trabalha para um gerente específico e que um gerente supervisiona um ou mais empregados.

Note que trocamos o atributo nome_agente do conjunto de entidades cliente pelo conjunto de relacionamentos agente_cliente, e o atributo gerente do conjunto de entidades empregado pelo conjunto de relacionamentos trabalha_para. Optamos por manter o conjunto de entidades empréstimo. O conjunto de relacionamentos agência_empréstimo e pagamento_empréstimo foi substituído, respectivamente, pelos atributos agência_origem e pagamento_empréstimo do conjunto de entidades empréstimo.”

Finalmente os autores demonstram, via Diagrama Entidade Relacionamento o que a descrição do sistema queria especificar.

“Deseja-se montar um sistema de informações para um determinado banco. Durante a etapa de levantamento de dados observou-se que:

- 1. Cada cliente do banco é identificado pelo número do seu seguro social, e possui ainda nome, rua e cidade, sendo atendido ainda por um empregado específico que cuida de seus negócios ou atua como um agente de empréstimos.*
- 2. Cada empregado do banco também é identificado por meio do número de seu seguro social, além de possuir o nome, o seu número do telefone, o número do seguro social do seu gerente, sua data de contratação e salário.*
- 3. Cada empregado pode ainda ter vários dependentes.*
- 4. Cada dependente de um empregado tem o seu nome.*
- 5. Cada conta é identificada através de um número tem um saldo e armazena a data do último acesso a ela. Caso a conta seja de Poupança ela possuirá ainda o taxa de juros. Caso a conta seja de movimento ela possuirá um limite de cheque especial.*
- 6. Cada conta pode pertencer a vários clientes.*
- 7. Todo empréstimo é identificado através de um número, e tem como atributos o montante e qual foi a agência que o originou.*
- 8. Um empréstimo pode ser obtido por mais de um cliente.*
- 9. Cada empréstimo possui várias parcelas.*
- 10. Cada parcela de um empréstimo é identificada por um número de pagamento tem a data do pagamento e o montante recebido.*
- 11. Cada agência é identificada por um nome único, está em uma cidade e tem um fundo financeiro.”*

Neste caso, na frase 2, foi omitido o campo tempo de trabalho, visto ser este um campo calculado que estaria infringindo a 3ª Forma Normal.

Para este exemplo foi respeitada integralmente a idéia original do sistema, não se questionando uma possível especialização em cliente e empregado e o porque da falta de identificação da agência na conta, além da necessidade de se possuir um número identificador de cada pagamento do empréstimo, visto esta ser apenas uma entidade fraca de empréstimo. Todas estas observações foram levantadas pelos grupos e os resultados foram muito próximos ao desejado pelo primeiro modelo, mesmo sendo este um enunciado muito menor em tamanho, porém mais detalhado em cada frase e organizado logicamente, o que facilitou a interpretação dos alunos. Uma falha geral da análise foi encontrada na frase de número 2, onde apenas um grupo conseguiu identificar o auto-relacionamento desejado.

3.5 – Confronto com as regras de Lista de Dependências

Como validação do sistema, abaixo seguem as explicações de como será garantida cada uma das regras definidas pelo item lista de dependências da técnica de Lista Diagrama.

A primeira regra que solicita que cada frase contenha apenas uma dependência multivalorada (DMV) ou várias dependências simples, desde que estas façam referência a um mesmo sujeito será alcançada através da interface de entrada das regras do sistema juntamente com a etapa de análise sintática;

A segunda regra solicita a numeração de cada frase, o que será realizado automaticamente no momento da entrada de dados no sistema, observando as indicações de especializações realizadas no campo Indicativo do Sujeito da Frase (a).

A terceira regra, que proíbe o uso de sinônimos para referenciar a um mesmo componente da lista de dependências exige um conhecimento contextual do sistema que está sendo proposto, pois, mesmo um item não sendo considerado sinônimo literal, ele pode perfeitamente estar representando um no momento em que o usuário o quis definir. Para

atingir esta regra será criada uma base de sinônimos que deverão ser confrontados para os sujeitos de cada frase. Também deverão ser analisados todos os sujeitos das frases que não contiverem atributos e os relacionamentos multivalorados onde os componentes não tenham sido referenciados nas regras do sistema. No caso dos atributos, onde pode existir uma dependência funcional simples, a análise é muito difícil, visto que os substantivos lá definidos podem ser atributos do sujeito desta frase, ou simplesmente estar fazendo referência à outro sujeito ou seu identificador. Nesta situação fica muito impreciso identificar um sinônimo baseado nos sujeitos já definidos, o que pode confundir o usuário.

A quarta regra que solicita que esteja claro cada componente de uma frase, na ordem Sujeito + Verbo + Predicado será alcançada através da interface no cadastramento das regras do negócio.

A quinta regra que solicita clareza na escolha do verbo-artigo que indicam a cardinalidade da dependência funcional será alcançada também através da interface, pois o usuário somente poderá indicar no campo Verbo de Ligação da Frase (e) algo que já esteja armazenado no banco de dados. Na etapa de sintaxe também deverá ser validada a concordância entre a dependência simples e itens no singular e na dependência multivalorada e itens no plural.

A sexta regra, que solicita que não sejam utilizados atributos não atômicos na especificação do sistema, será alcançada na etapa de análise semântica, onde o agente validará os substantivos informados como atributos em uma tabela própria de atributos reconhecidamente não atômicos.

A sétima regra, que solicita a ordenação das frases conforme sua importância para o sistema, será alcançada na etapa de análise semântica, onde serão analisados os sujeitos de cada frase (b) ou os identificadores (d) e verificado a existência destes no campo de atributos (f) em outras frases.

A oitava e última que solicita a indicação das dependências facultativas será alcançada através do campo Identificador do sujeito (a) na interface de entrada e também pela análise semântica do campo de atributos (f).

4. – Protótipo do Sistema

4.1 – Definição da Base de dados de suporte.

Inicialmente, para efeito de testes e evolução do sistema foi definida uma base de dados de suporte para que o software possa ser executado. Como protótipo inicial foi utilizado o SGBD MS-ACCESS onde foi definida a seguinte estrutura:

4.1.1 – Tabelas

Sistema – Tabela de cadastro básico com os dados de identificação do sistema que será analisado.

Frase – Tabela de armazenamento inicial das frase que o projetista irá cadastrar para análise pelo sistema.

PalavrasSistema – tabela onde o agente de análise morfológica armazenará as palavras identificadas, classificando-as.

Palavra – Dicionário de palavras conhecidas para suportar o analisador morfológico.

TipoPalavra – Relacionamento entre TIPO e PALAVRA que classifica a palavra. Este relacionamento é necessário para que seja possível decisão, por parte do usuário, que qual tipo da palavra está sendo utilizado caso existam dois cadastros da mesma.

Tipo – Tabela de tipos de palavras possíveis para o sistema.

4.1.2 – Relacionamentos

Os relacionamentos definidos servem para que os agentes possam recuperar as informações armazenadas mais rapidamente, utilizando-se para isso dos recursos existentes no modelo relacional.

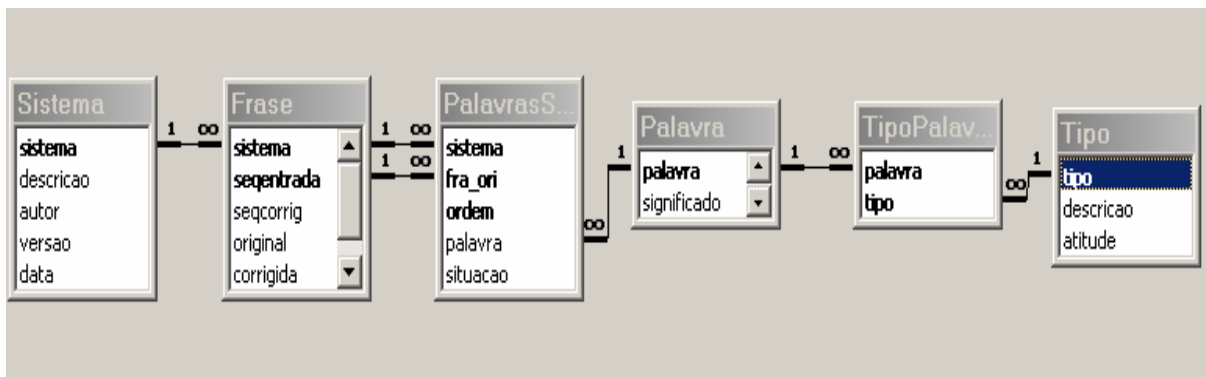


Figura 26 - Modelo Relacional da Base de Dados de Suporte

5. - Projeto

5.1 Soluções Propostas Para o Sistema

Desenvolver uma ferramenta contendo:

- a) Cadastro do sistema, regras, palavras, sinônimo, palavras não atômicas. Estes cadastros conterão todas as informações necessárias para o aluno e permitirão a inclusão, alteração e exclusão de dados.
- b) Validação das regras, satisfazendo o passo 1, criação de lista de dependência, dentro do modelo lista-diagrama citadas anteriormente na revisão bibliográfica.

5.2 Documentação da Análise

Para documentação de análise do sistema será utilizada a modelagem através de UML - The Unified Modeling Language – pelo motivo desta apresentar-se clara e simples nas definições das classes de objetos e, principalmente, os métodos envolvidos para definição dos agentes que suportarão este sistema. Como notação básica será utilizada a aplicada na ferramenta case Rational Rose.

5.2.1 Lista de Use Cases

Como definição inicial serão tratados os seguintes casos de uso (Use Cases):

1. cadastrarSistema
2. cadastrarPalavra
3. cadastrarRegra
4. cadastrarSinonimo
5. cadastrarPalFixa
6. cadastrarPalNaoAtomica
7. gerarRelPalFixa
8. gerarRelPalNaoAtomica
9. gerarRelatorioSistema
10. gerarRelRegraOriginal
11. gerarRelatorioPalavra
12. gerarRelatorioRegra

5.2.2 Lista de eventos

Nº	Use Case	Descrição	Evento	Resposta
01	CadastrarSistema	Aluno solicita cadastrar sistema	dadosSistema	Msg01
02	CadastrarPalavra	Aluno solicita cadastrar palavra	dadosPalavra	Msg02
03	CadastrarRegra	Aluno solicita cadastrar regra	dadosRegra	Msg03
04	CadastrarSinonimo	Professor solicita cadastrar sinônimo	dadosSinonimo	Msg04
05	CadastrarPalFixa	Professor solicita cadastrar palavra fixa	dadosPalavraFixa	Msg05
06	CadastrarPalNaoAtomica	Professor solicita cadastrar palavra não atômica	dadosPalavraNao Atomica	Msg06
07	GerarRelPalFixa	Professor solicita relatório de palavra fixa	solicitaRelatorio	relPalFixa
08	GerarRelPalNaoAtomica	Professor solicita relatório de palavra não atômica	solicitaRelatorio	relPalNaoAtomica
09	GerarRelatorioSistema	Professor solicita relatório de sistema	solicitaRelatorio	relSistema
10	GerarRelRegraOriginal	Professor solicita relatório de regra original	solicitaRelatorio	relRegraOriginal
11	GerarRelatorioPalavra	Professor solicita relatório de palavra	solicitaRelatorio	relPalavra
12	GerarRelatorioRegra	Professor solicita relatório de regra	solicitaRelatorio	relRegra

Observações:

Significado das Mensagens de Erros

Msg01 - Sistema Cadastrado / Sistema Alterado / Sistema Excluído / Sistema já existe /

Não será possível excluir o sistema existem regras associadas a ele

Msg02 - Palavra Cadastrada / Palavra Alterada / Palavra Excluída / Palavra já existe

Msg03 - Regra Cadastrada / Regra Alterada com sucesso / Regra Excluída / Existe erro na frase, deseja continuar?

Msg04 - Sinônimo Cadastrado / Sinônimo Alterado com sucesso / Sinônimo Excluído

Msg05 - Palavra Cadastrada / Palavra Alterada com sucesso / Palavra Excluída / Palavra já existe

Msg06 - Palavra Cadastrada / Palavra Alterada com sucesso / Palavra Excluída / Palavra já existe

5.2.3 - Diagramas de Use Case

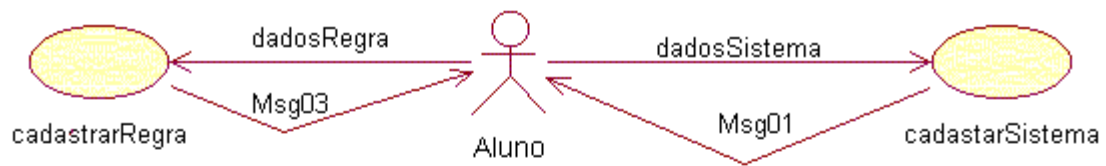


Figura 27 – Use Case Principal do Sistema

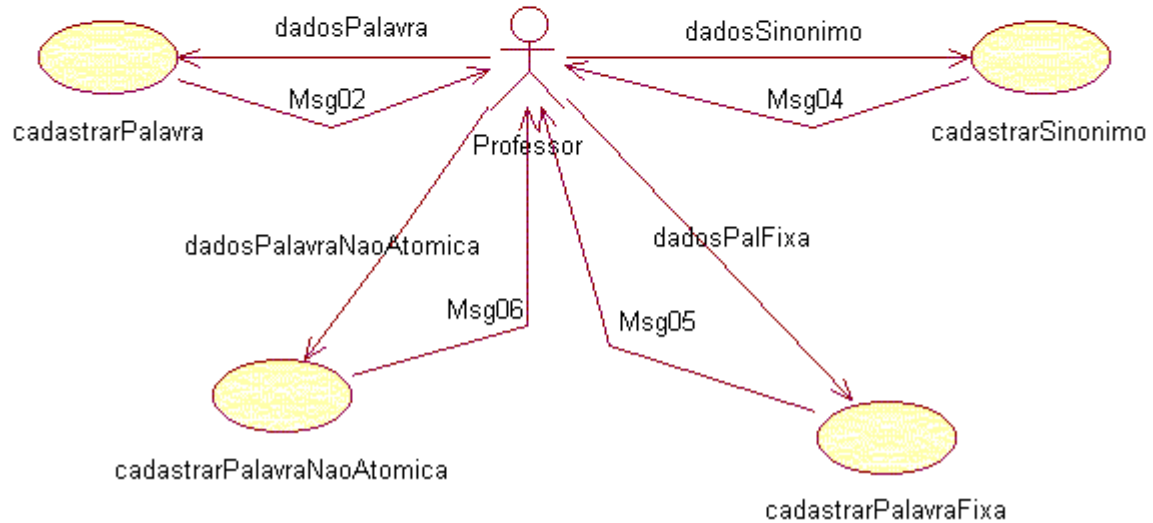


Figura 28 – Use Case de Apoio ao Sistema

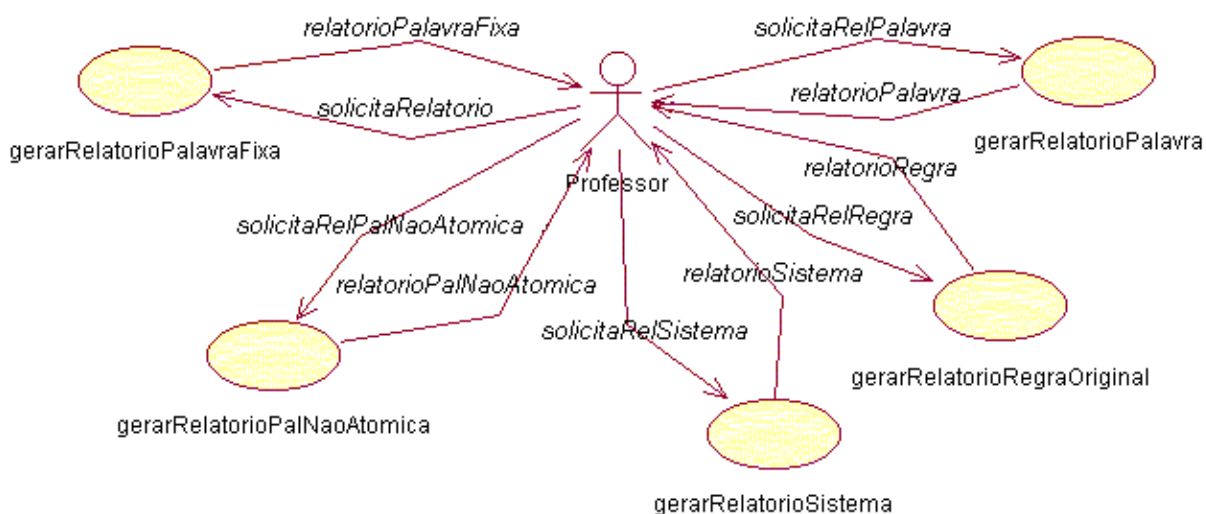


Figura 29 – Use Case dos relatórios do Sistema

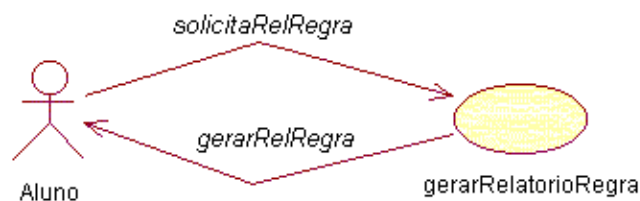


Figura 30 – Use Case do Relatório de Regras do Sistema

5.2.4 Use Cases

Número: 01

Use Case: cadastrarSistema

Descrição: Este Use Case trata do cadastro de um sistema.

Ator: Aluno

Curso Normal:

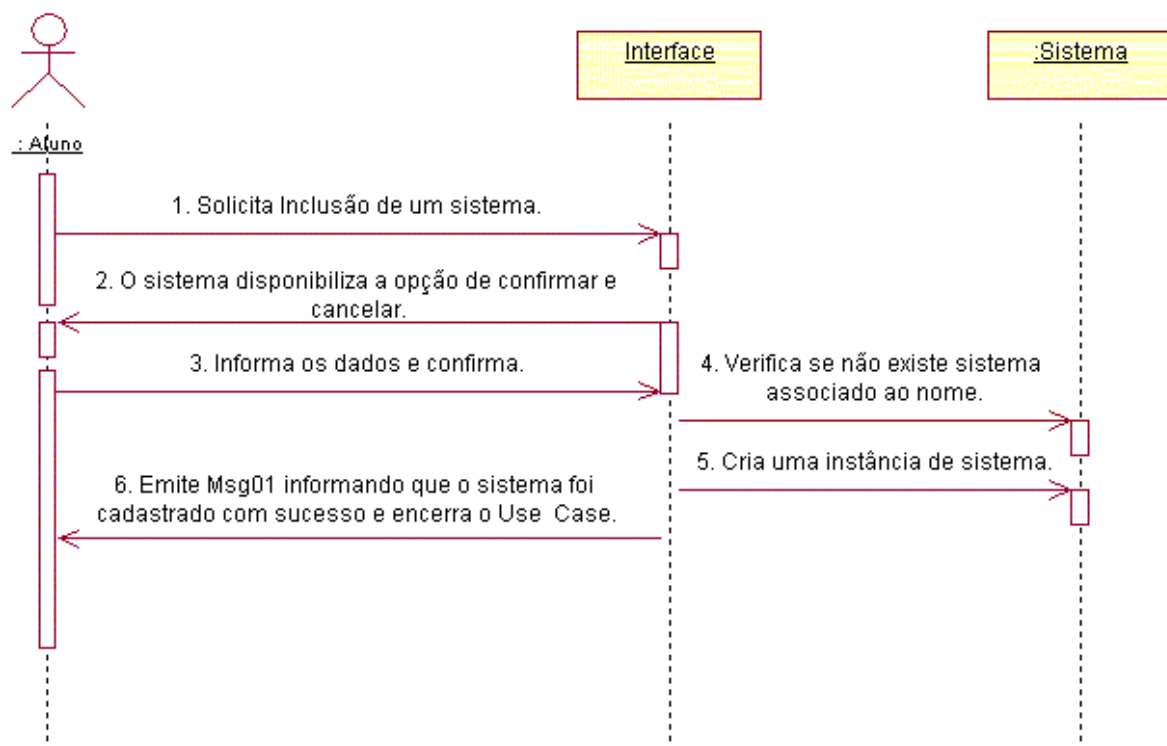


Figura 30 - Use Case 01 – cadastrarSistema – Curso Normal

1. Aluno solicita a inclusão de um sistema.
2. O sistema disponibiliza a opção de confirmar e cancelar.
3. O aluno informa os dados e confirma.

4. O sistema verifica se não existe sistema associado ao nome.
5. O sistema cria uma instância de sistema.
6. O sistema emite Msg01 informando que o sistema foi cadastrado e encerra o Use Case

Diagrama de Colaboração

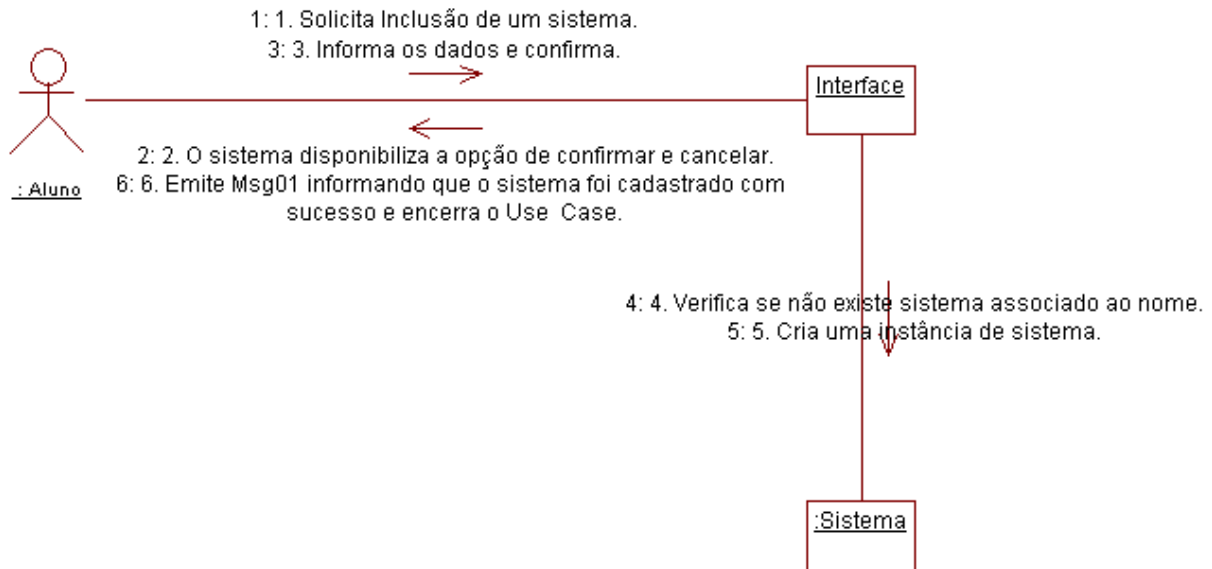


Figura 31 – Diagrama de Colaboração – cadastrarSistema – Curso Normal

Curso Alternativo:

Alternativo 1. O aluno solicita a busca de um sistema.

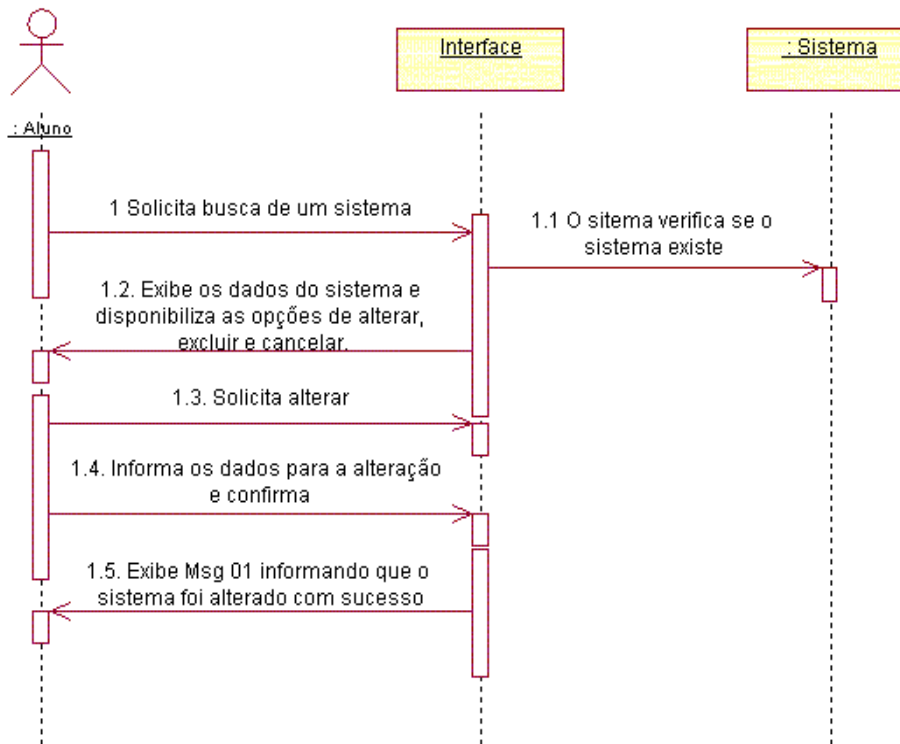


Figura 32 - Use Case 01 – cadastrarSistema – Curso Alternativo 1

1.1 O sistema verifica se sistema existe.

1.2. O sistema exibe os dados do sistema e disponibiliza as opções de alterar, excluir e cancelar.

1.3. O aluno solicita alterar.

1.4. O aluno informa os dados para a alteração e confirma.

1.5. O sistema emite Msg01 informando que o sistema foi alterado e encerra o Use Case.

Diagrama de Colaboração

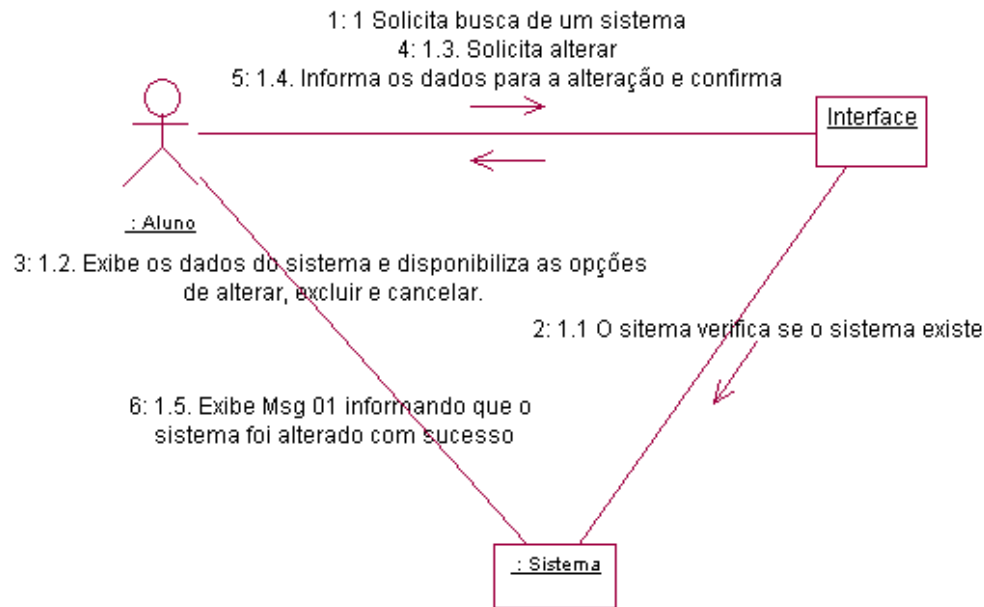


Figura 33 – Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1

Alternativo 3. O aluno solicita cancelar.

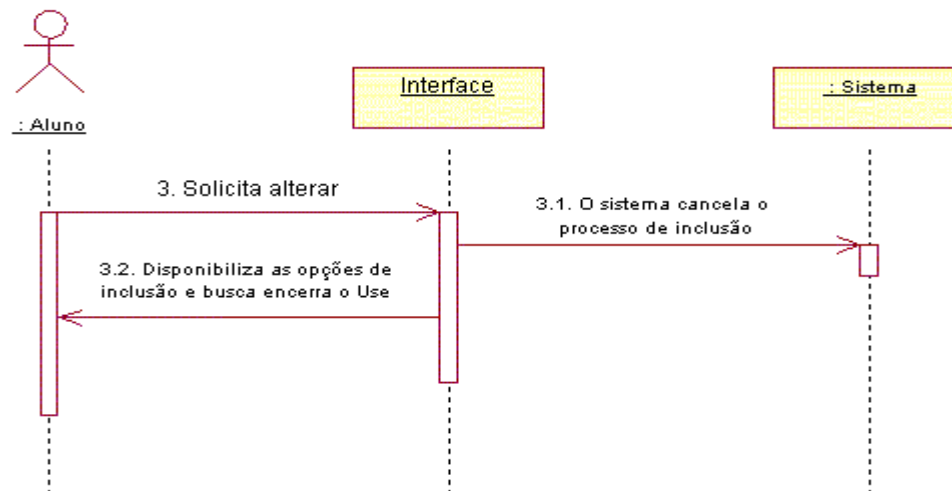


Figura 34 - Use Case 01 – cadastrarSistema – Curso Alternativo 3

3.1. O sistema cancela o processo de inclusão.

3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

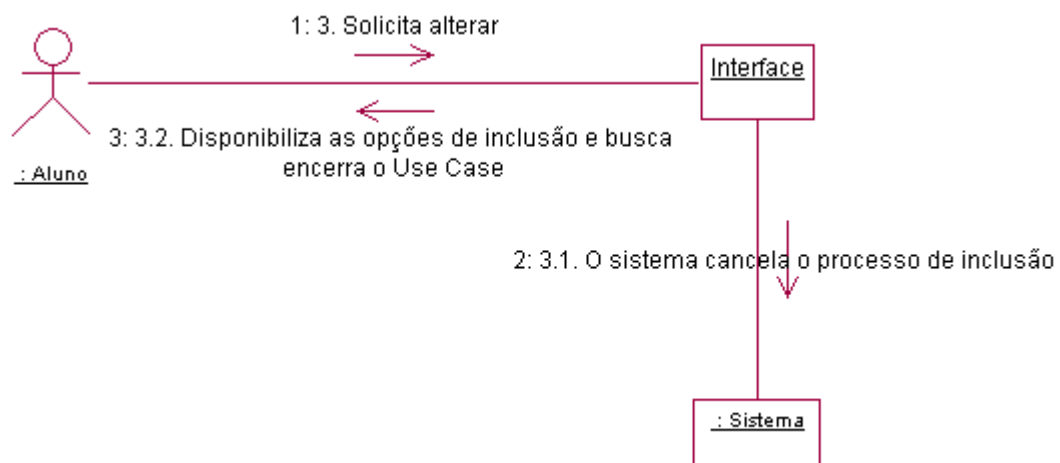


Figura 35 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 3

Alternativo 4. O sistema verifica que existe sistema associado ao nome.

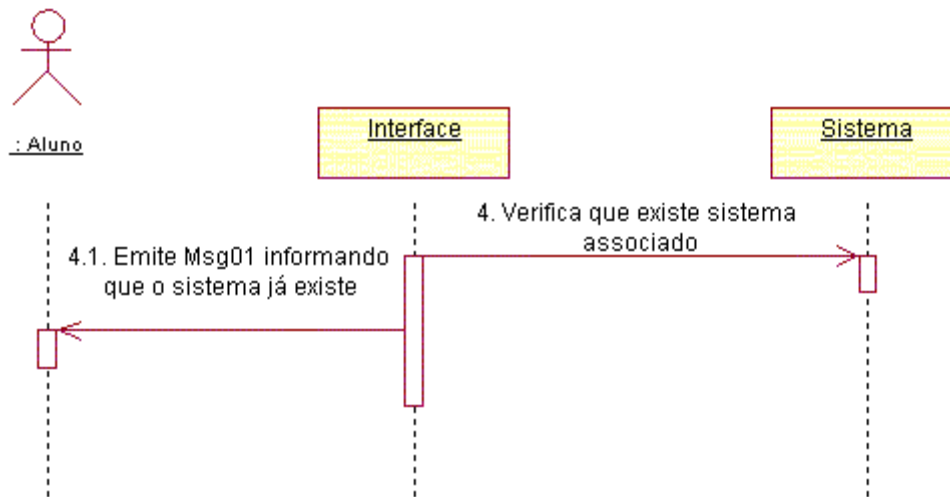


Figura 36 - Use Case 01 – cadastrarSistema – Curso Alternativo 4

4.1. O sistema emite Msg01 informando que o sistema já existe.

Diagrama de Colaboração

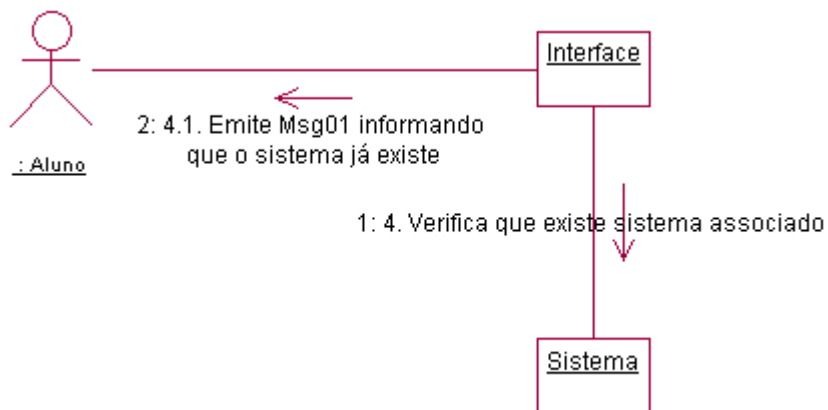


Figura 37 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 4

Alternativo 1.1. O sistema verifica que não existe sistema.

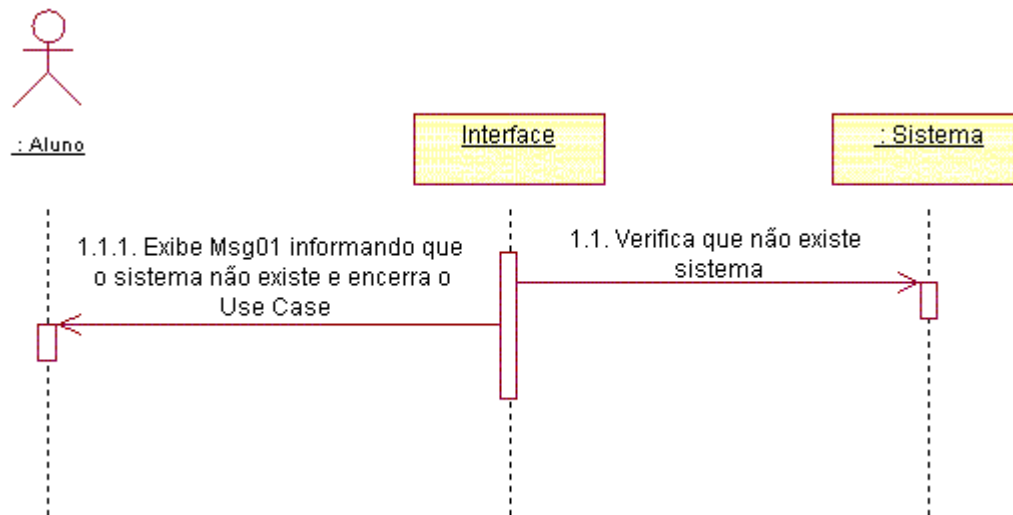


Figura 38 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.1

1.1.1 O sistema emite Msg01 informando que o sistema não existe e encerra o Use case.

Diagrama de colaboração

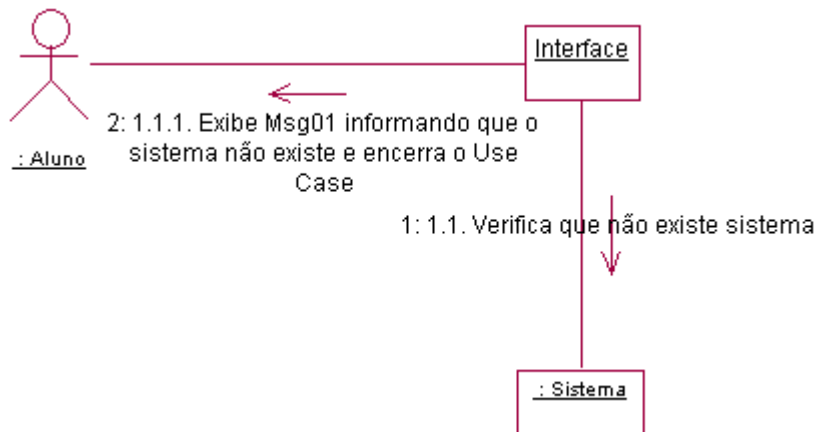


Figura 39 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.1

Alternativo 1.3. O aluno escolhe a opção de excluir.

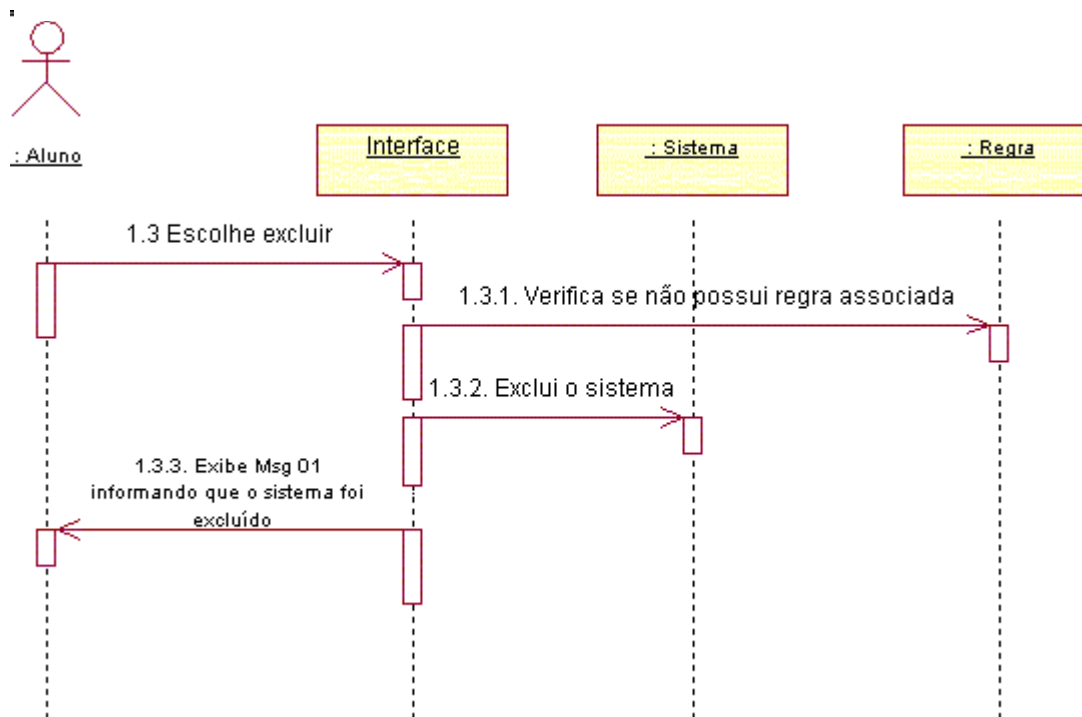


Figura 40 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3

1.3.1. O sistema verifica se não existem regras associadas ao sistema.

1.3.2. O sistema exclui o sistema.

1.3.3. O sistema exibe Msg01 informando que o sistema foi excluído e encerra o Use Case.

Diagrama de Colaboração.

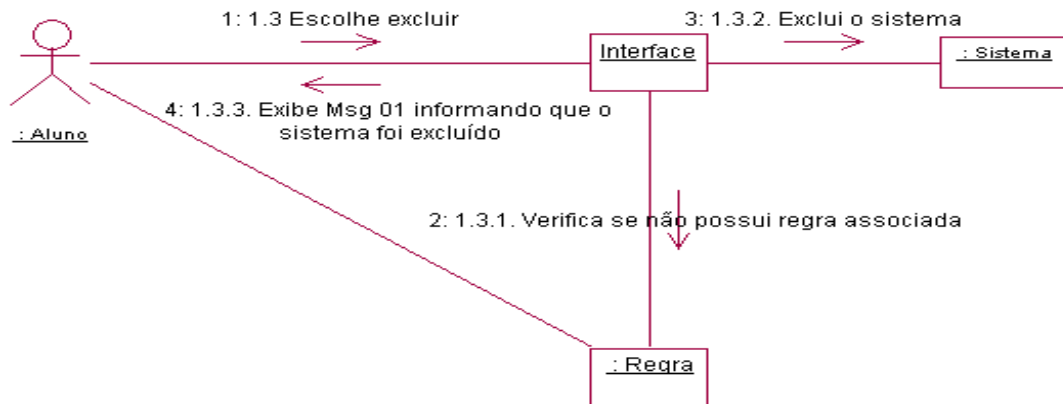


Figura 41 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3

Alternativo 1.3. O aluno escolhe a opção de cancelar.

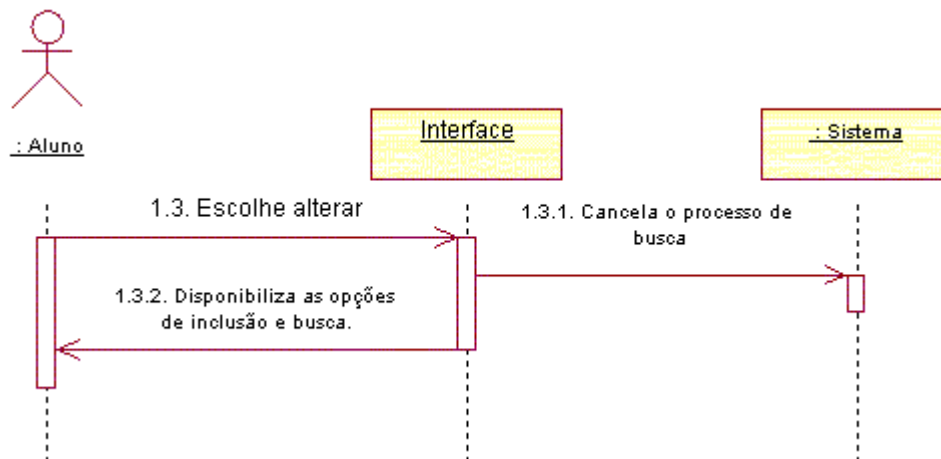


Figura 42 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3

1.3.1. O sistema cancela o processo de busca.

1.3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

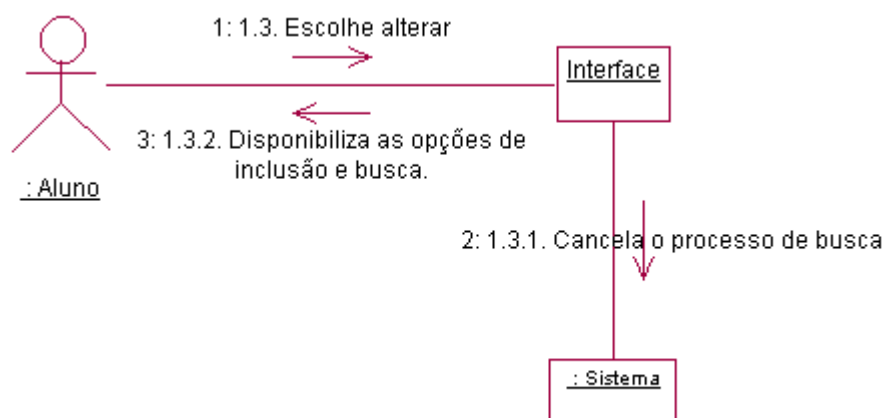


Figura 43 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3

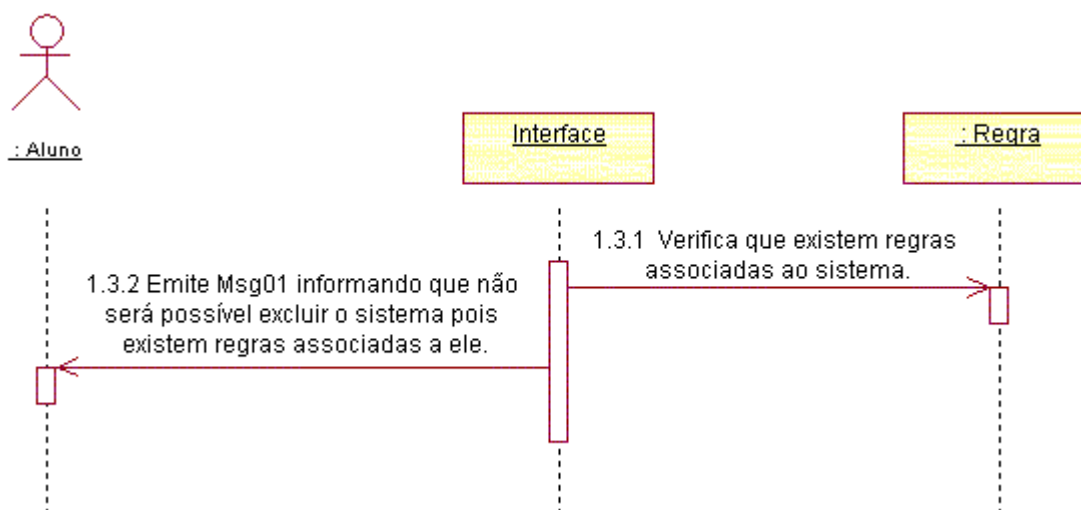


Figura 44 - Use Case 01 – cadastrarSistema – Curso Alternativo 1.3.1

Alternativo 1.3.1 O sistema verifica que existem regras associadas ao sistema.

1.3.2 O sistema emite Msg01 informando que não será possível excluir o sistema pois existem regras associadas a ele.

Diagrama de Colaboração

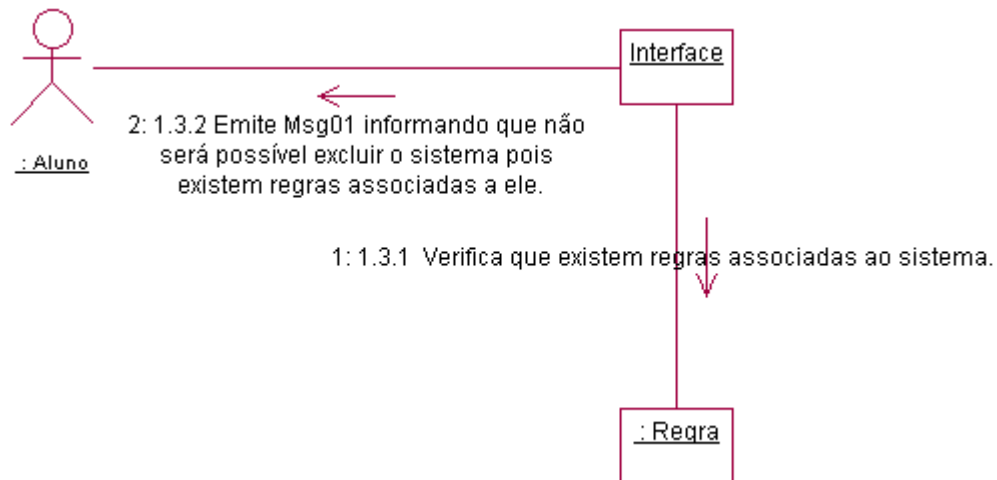


Figura 45 - Diagrama de Colaboração – cadastrarSistema – Curso Alternativo 1.3.1

Número: 02

Use Case: cadastrarPalavra

Descrição: Este Use Case trata do cadastro de palavras.

Ator: Professor

Curso Normal:

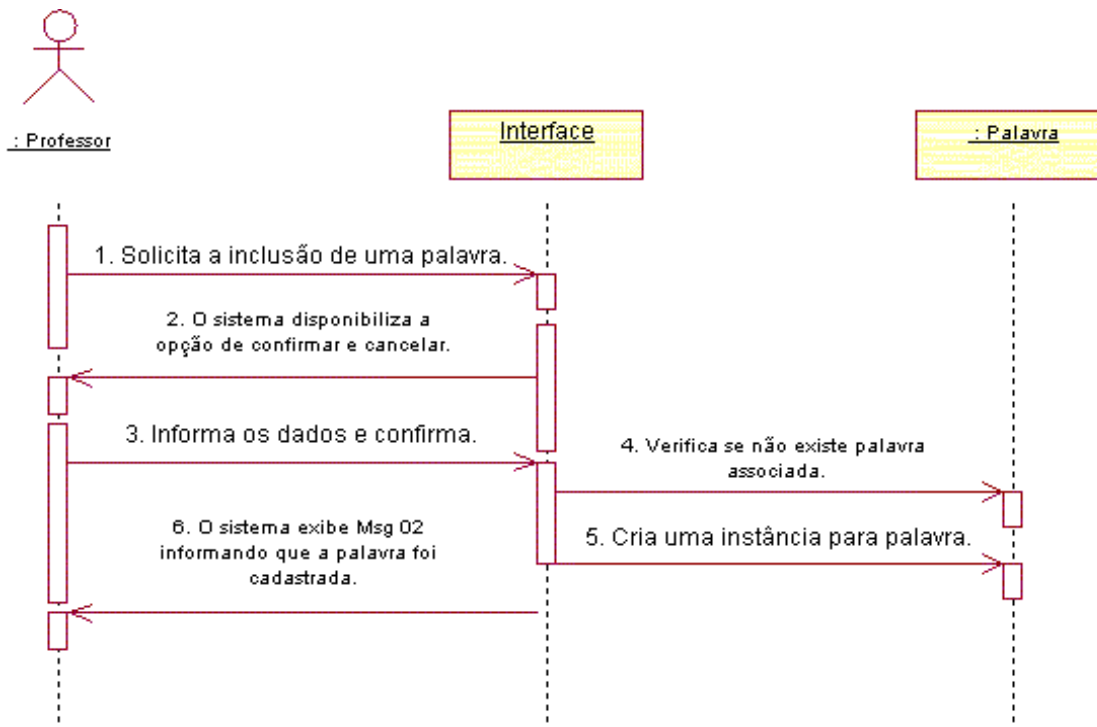


Figura 46 - Use Case 02 – cadastrarPalavra – Curso Normal

1. Professor solicita a inclusão de uma palavra.
2. O sistema disponibiliza a opção de confirmar e cancelar.
3. O professor informa os dados e confirma.
4. O sistema verifica se não existe palavra associada.
5. O sistema cria uma instância de palavra.
6. O sistema emite Msg02 informando que a palavra foi cadastrada e encerra o Use Case

Diagrama de Colaboração

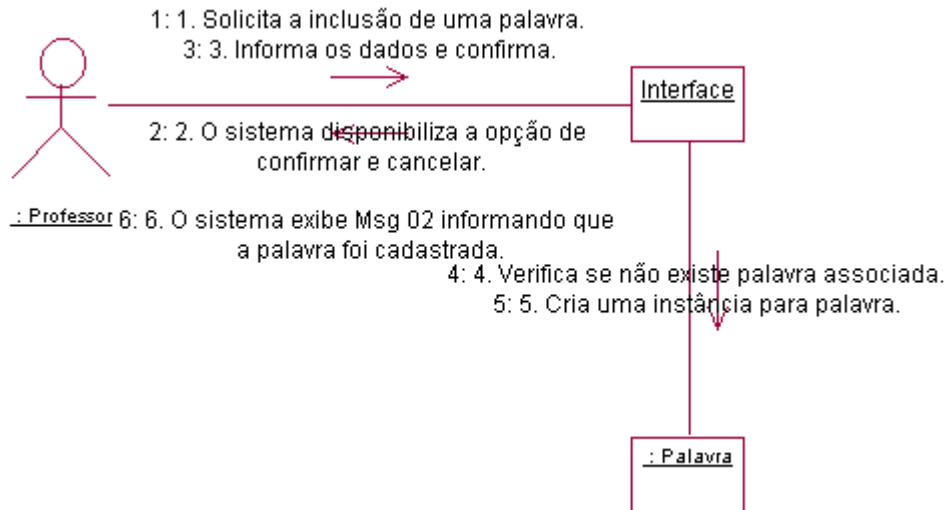


Figura 47 – Diagrama de Colaboração – cadastrarPalavra – Curso Normal

Curso Alternativo:

Alternativo 1. O professor solicita a busca de uma palavra.

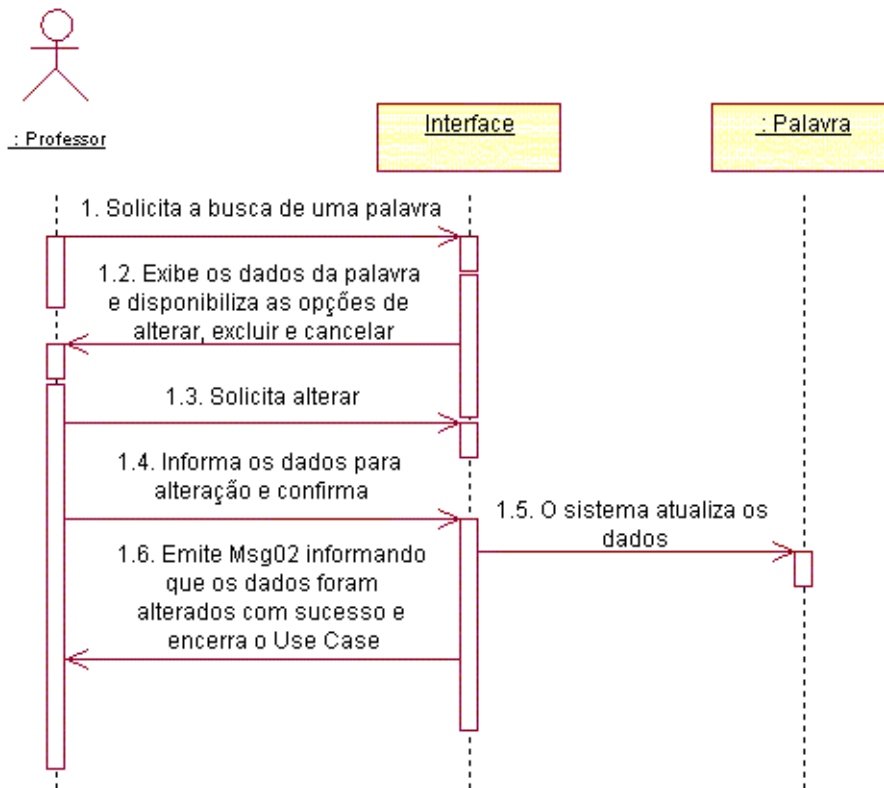


Figura 48 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1

- 1.2. O sistema exibe os dados da palavra e disponibiliza as opções de alterar, excluir e cancelar.
- 1.3. O professor solicita alterar.
- 1.4. O professor informa os dados para a alteração da palavra.
- 1.5. O sistema atualiza os dados.
- 1.6. O sistema emite Msg02 informando que a palavra foi alterada e encerra o Use Case.

Diagrama de Colaboração

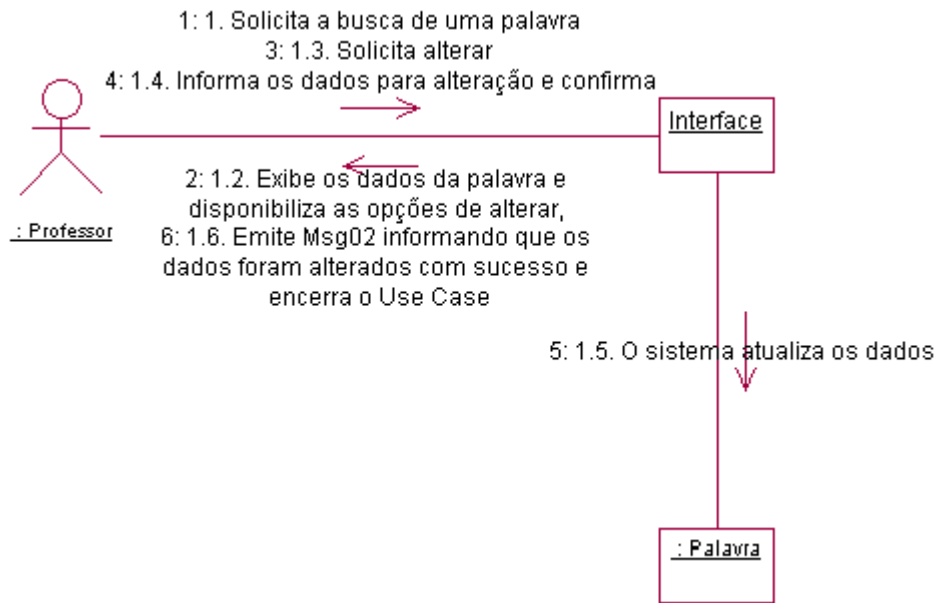


Figura 49 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1

Alternativo 3. O professor solicita cancelar.

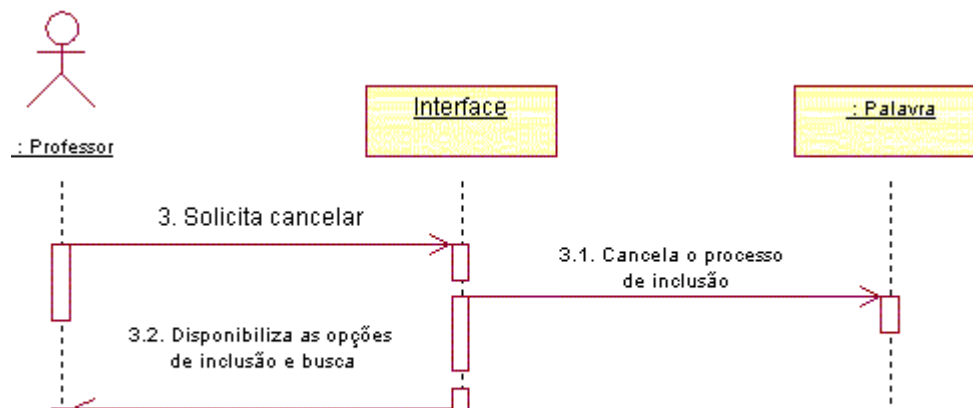


Figura 50 - Use Case 02 – cadastrarPalavra – Curso Alternativo 3

3.1. O sistema cancela o processo de inclusão.

3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

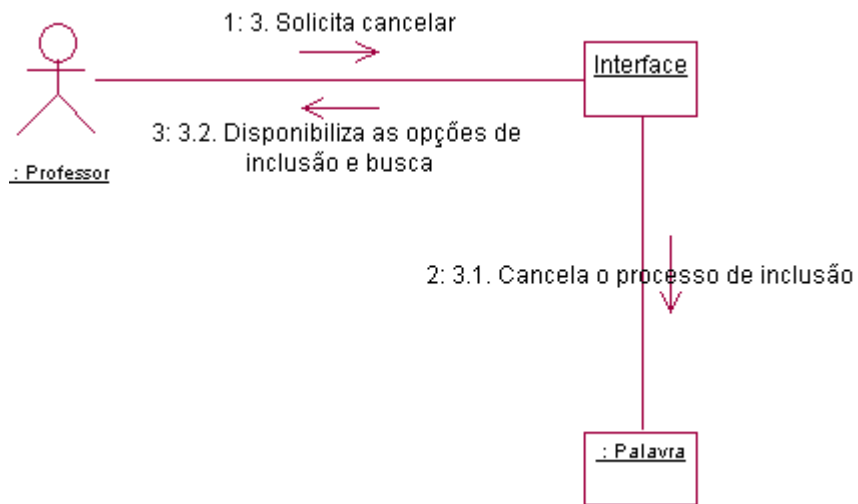


Figura 51 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 3

Alternativo 4. O sistema verifica que existe palavra associada.

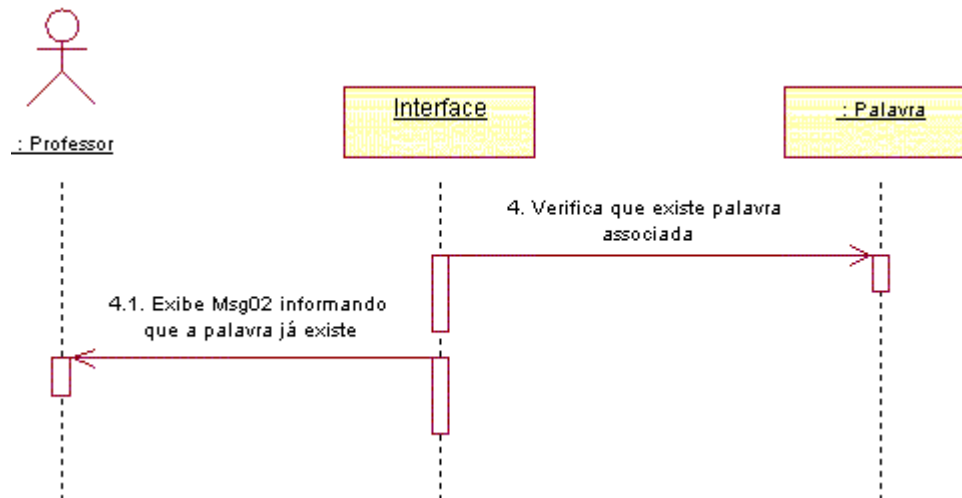


Figura 52 - Use Case 02 – cadastrarPalavra – Curso Alternativo 4

4.1. O sistema emite Msg02 informando que a palavra já existe.

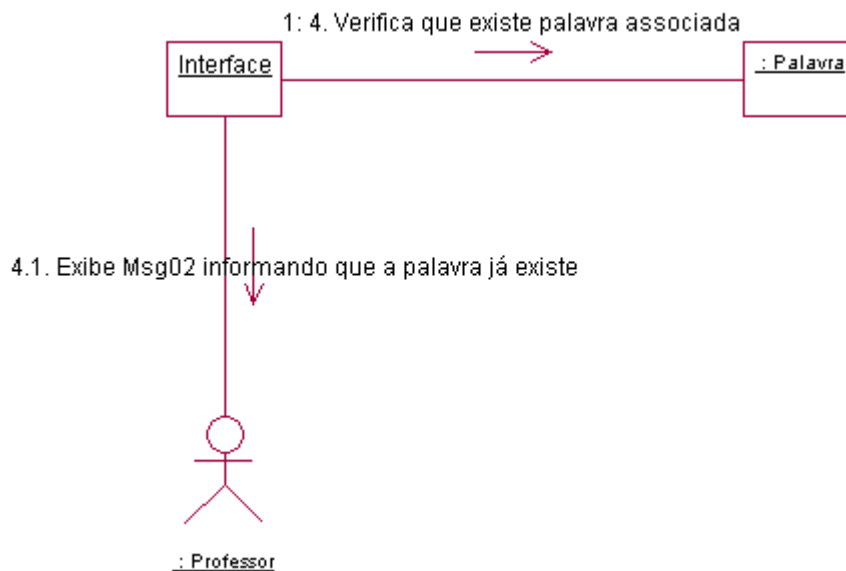


Figura 53 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 4

Alternativo 1.3. O professor escolhe a opção de excluir.

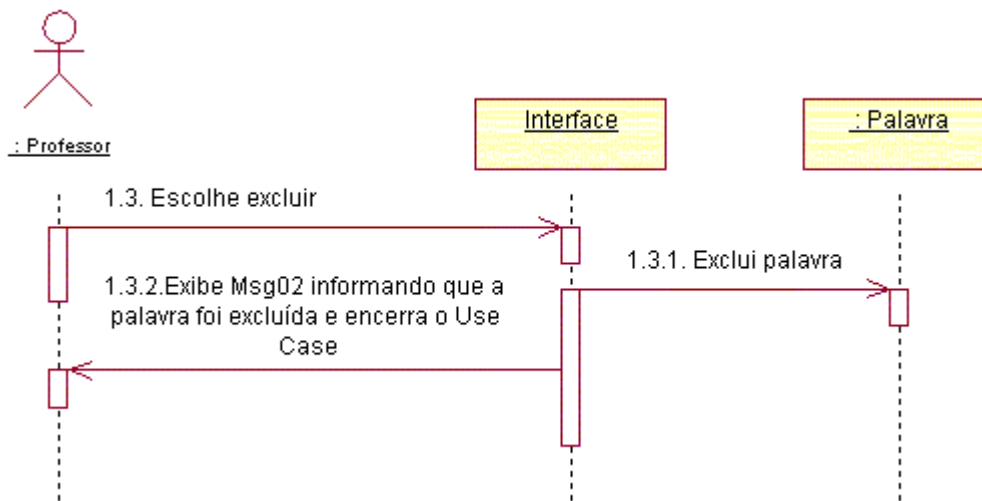


Figura 54 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1.3

1.3.1. O sistema exclui a palavra.

1.3.2. O sistema exibe Msg02 informando que a palavra foi excluída e encerra o Use Case.

Diagrama de Colaboração

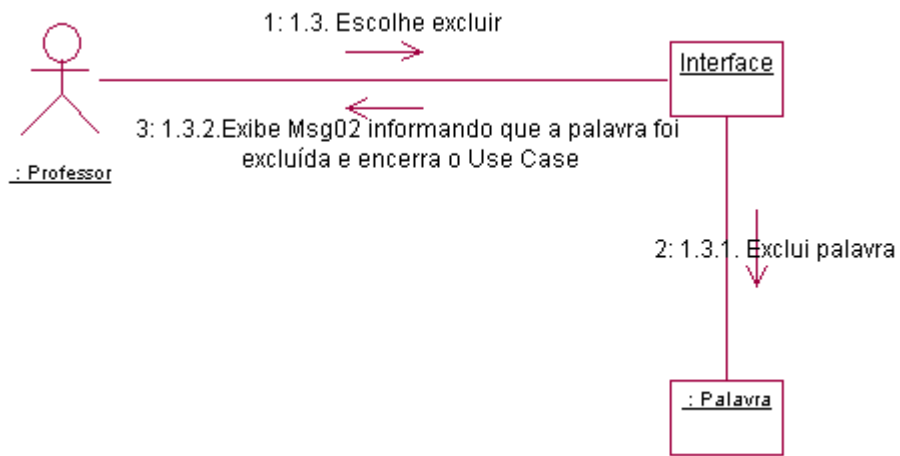


Figura 55 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1.3

Alternativo 1.3. O professor escolhe a opção de cancelar.

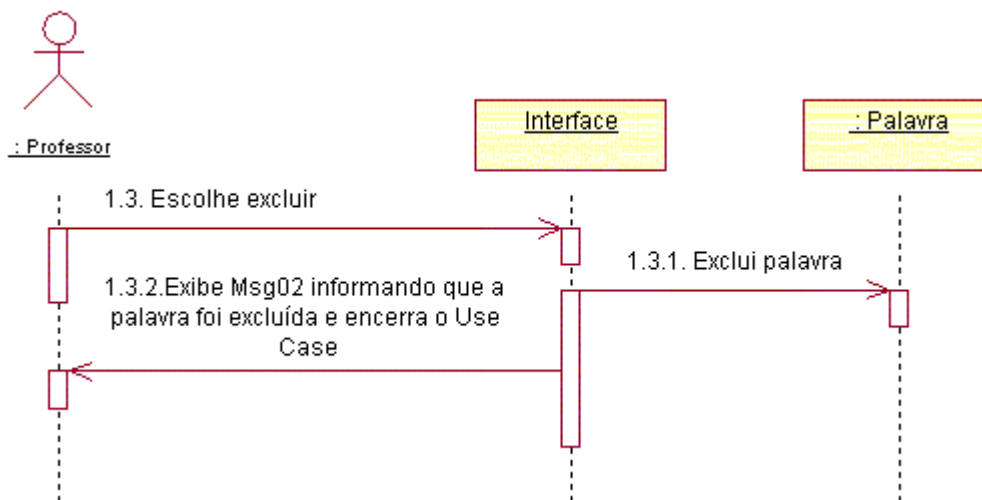


Figura 56 - Use Case 02 – cadastrarPalavra – Curso Alternativo 1.3

1.3.1. O sistema cancela o processo de busca.

1.3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

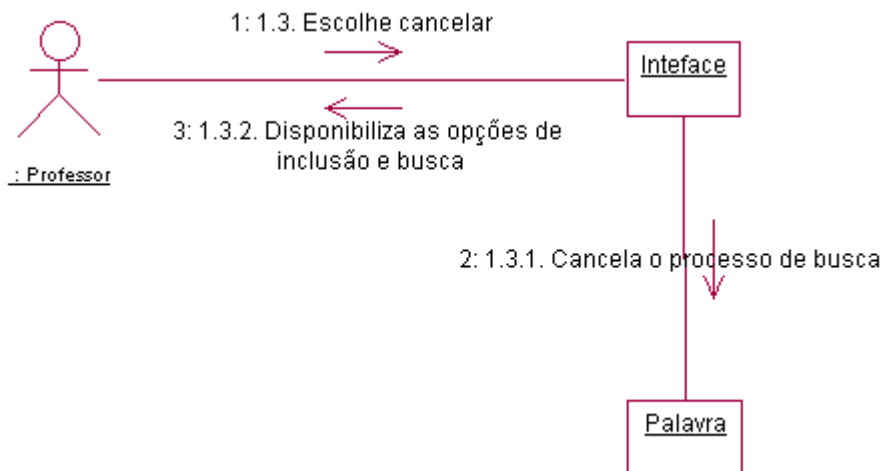


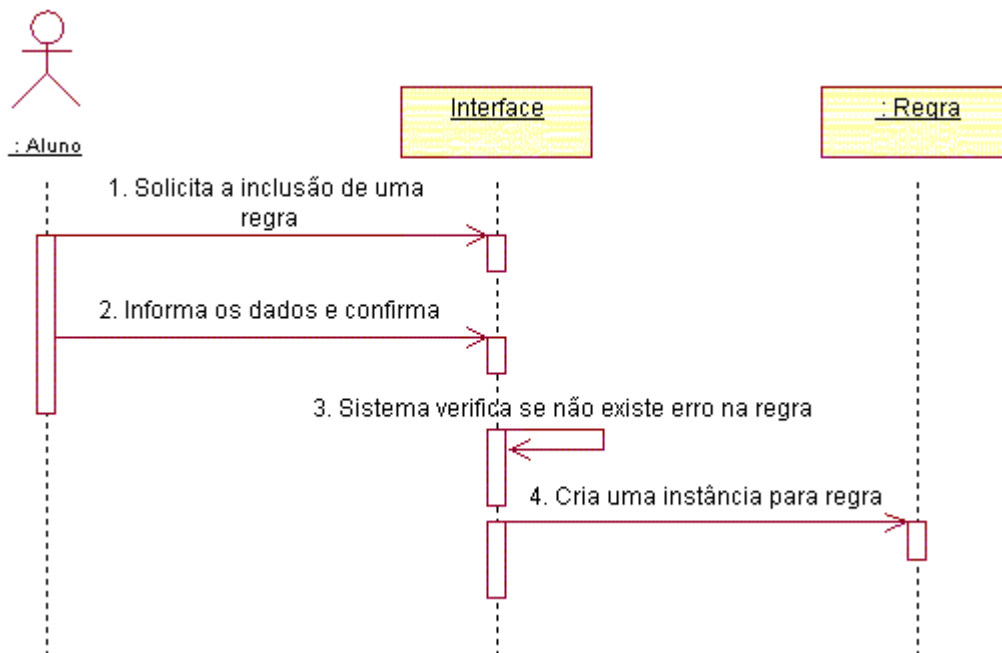
Figura 57 – Diagrama de Colaboração – cadastrarPalavra – Curso Alternativo 1.3

Número: 03

Use Case: cadastrarRegra

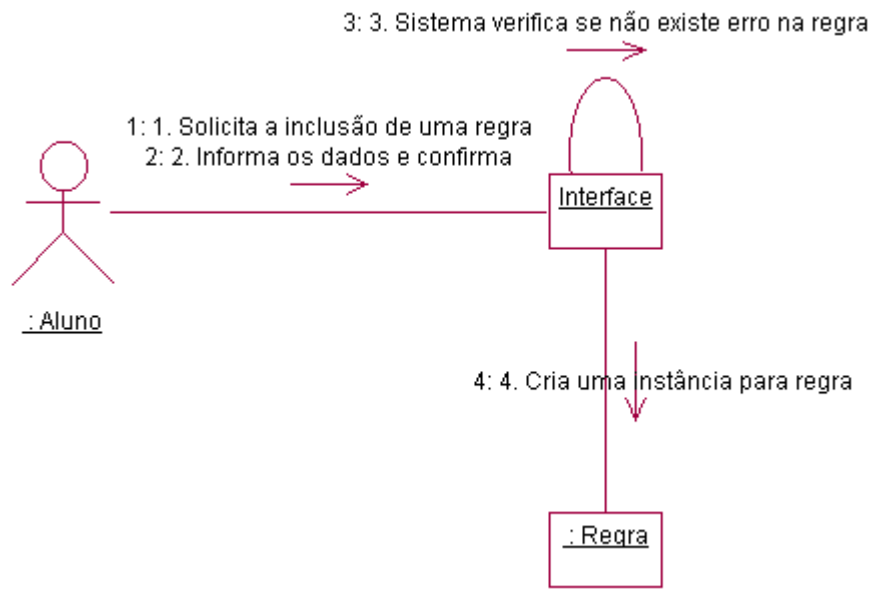
Descrição: Este Use Case trata do cadastro de palavras.

Ator: Aluno

Curso Normal:**Figura 58 - Use Case 03 – cadastrarRegra – Curso Normal**

1. Aluno solicita cadastrar regra.
2. O aluno entra com os dados da regra e confirma.
3. O sistema verifica se não existe nenhum erro na regra.
4. O sistema cria uma instância de regra e encerra o Use Case

Diagrama de Colaboração

**Figura 59 – Diagrama de Colaboração – cadastrarRegra – Curso Normal****Curso Alternativo:**

Alternativo 1. O aluno solicita alterar regra.

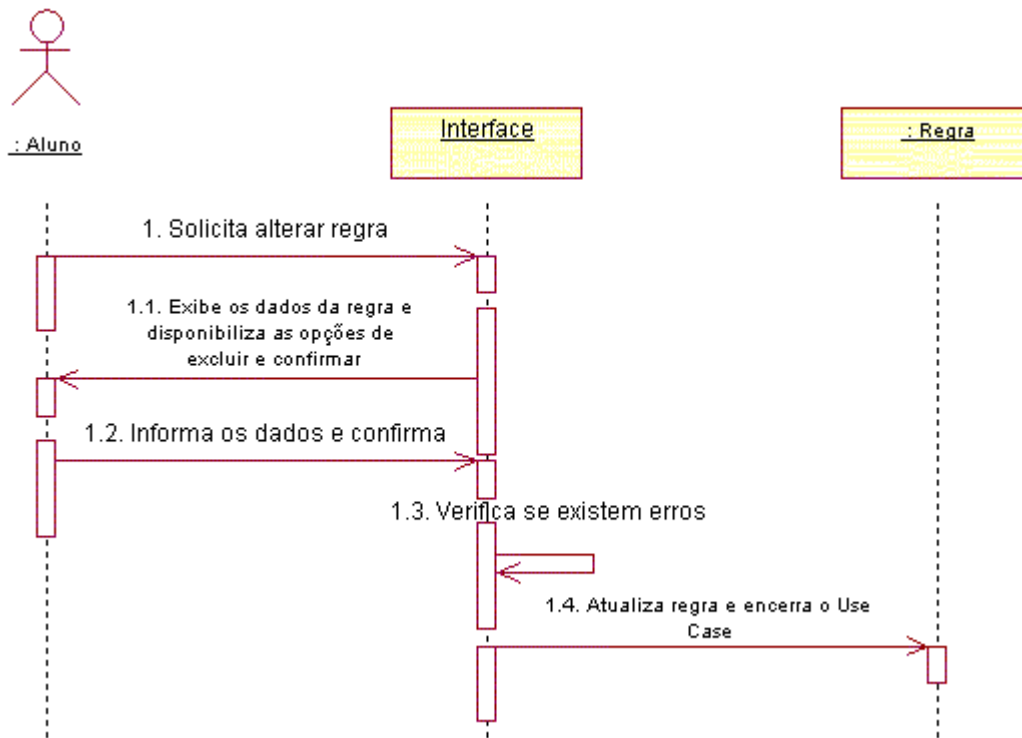


Figura 60 - Use Case 03 – cadastrarRegra – Curso Alternativo 1

1.1. O sistema exibe os dados da regra e disponibiliza as opções de excluir e confirmar.

1.2. O aluno informa os dados e confirma.

1.3. O sistema verifica se existem erros na frase e encerra o Use Case.

Diagrama de Colaboração

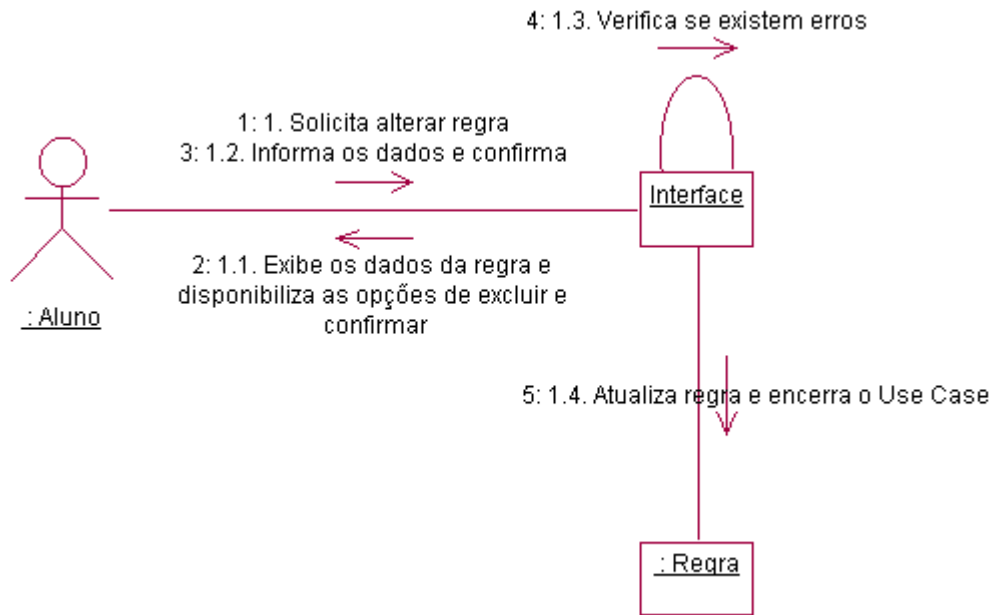


Figura 61 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 1

Alternativo 3. O sistema encontra erros na frase.

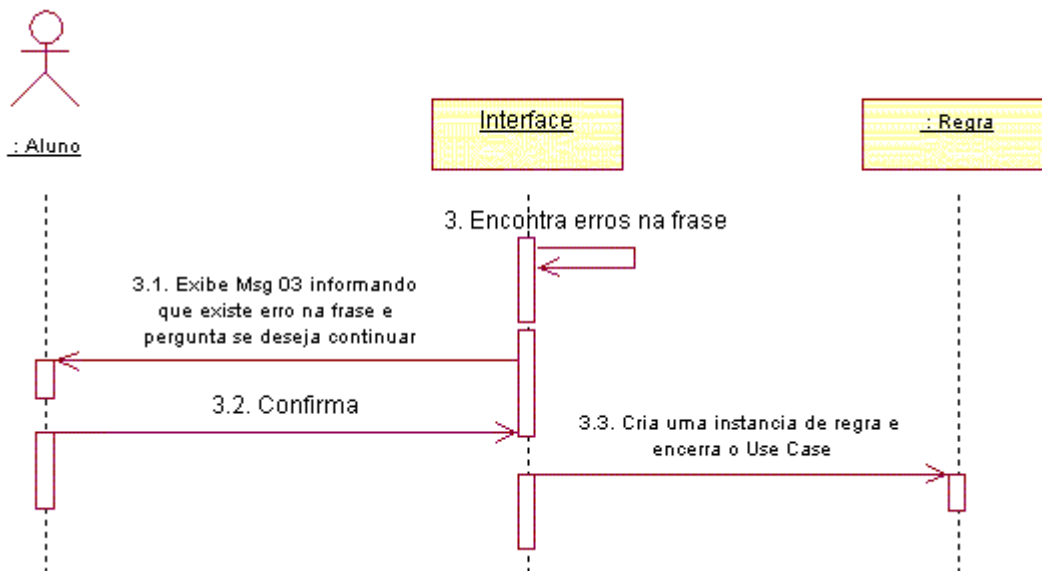
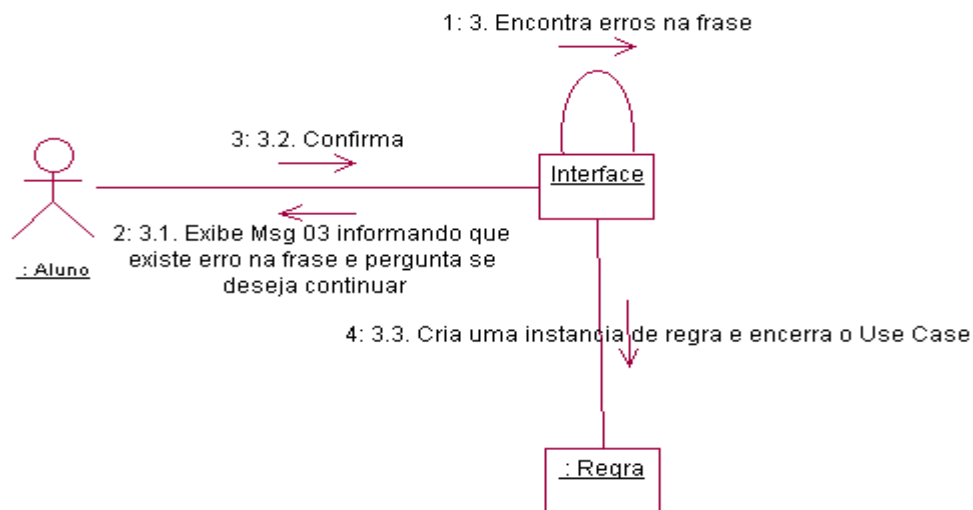


Figura 62 - Use Case 03 – cadastrarRegra – Curso Alternativo 3

- 3.1. O sistema exibe Msg03 informando que existe erro na frase e pergunta se deseja continuar.
- 3.2. O aluno confirma.
- 3.3. O sistema cria uma instancia de regra e encerra o Use Case

Diagrama de Colaboração**Figura 63 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 3**

Alternativo 3.2 O aluno não confirma.

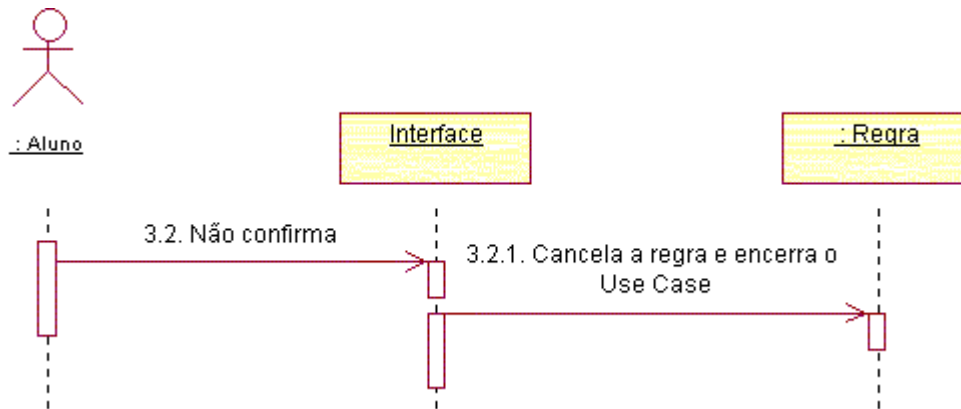


Figura 64 - Use Case 03 – cadastrarRegra – Curso Alternativo 3.2

3.2.1. O sistema cancela a regra e encerra o Use Case.

Diagrama de Colaboração

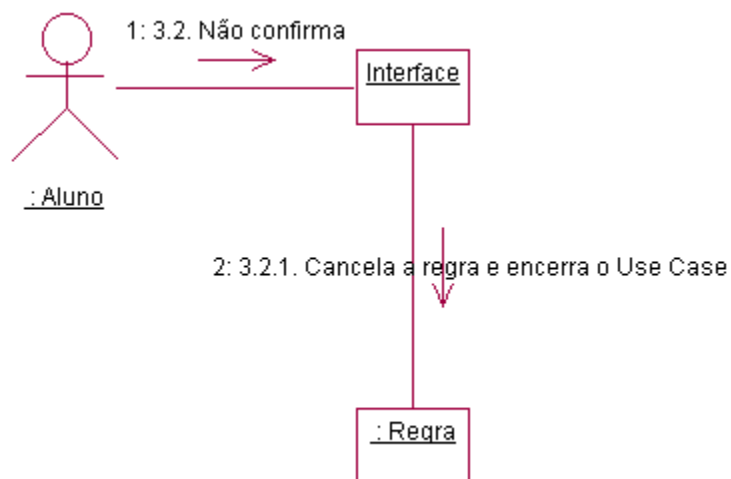


Figura 65 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 3.2

Alternativo 1.2. O aluno escolhe a opção de excluir

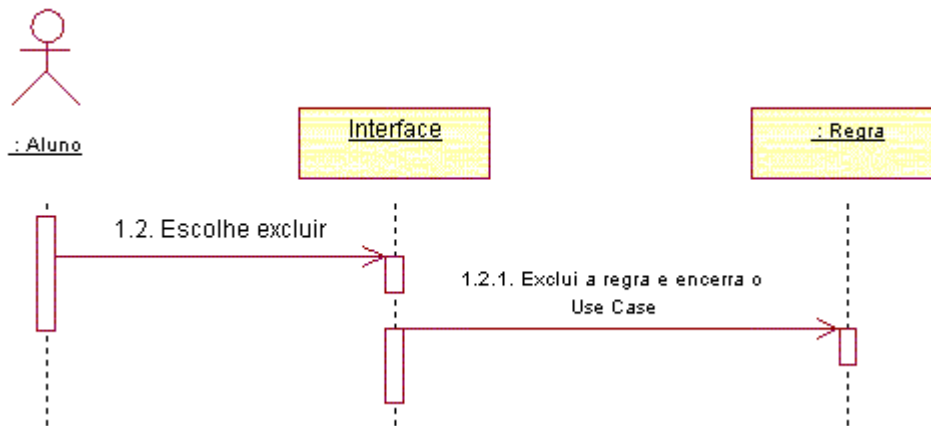


Figura 66 - Use Case 03 – cadastrarRegra – Curso Alternativo 1.2

1.2.1. O sistema exclui a regra e encerra o Use Case.

Diagrama de Colaboração

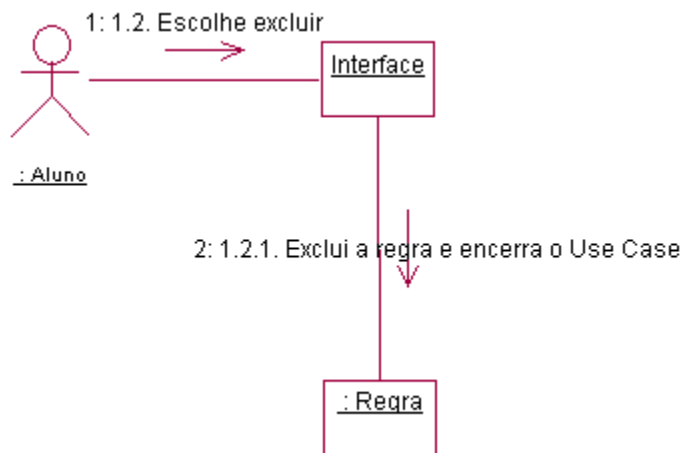


Figura 67 – Diagrama de Colaboração – cadastrarRegra – Curso Alternativo 1.2

Número: 04

Use Case: cadastrarSinonimo

Descrição: Este Use Case trata do cadastro de sinônimo de palavras.

Ator: Aluno

Curso Normal:

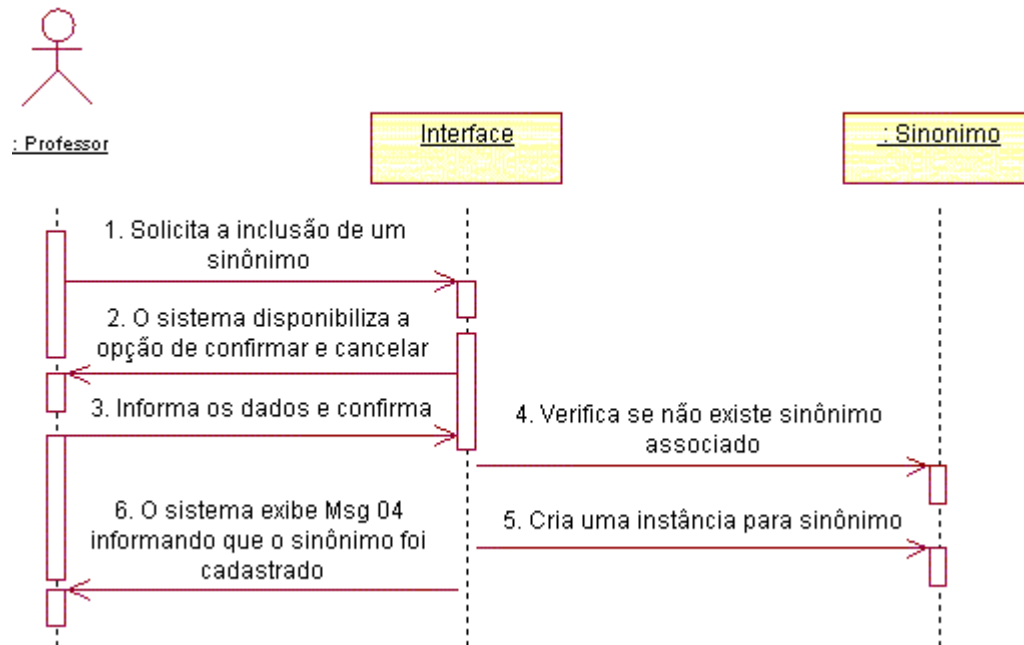


Figura 68 - Use Case 04 – cadastrarSinonimo – Curso Normal

1. Professor solicita a inclusão de um sinônimo.
2. O sistema disponibiliza a opção de confirmar e cancelar.
3. O professor informa os dados e confirma.
4. O sistema verifica se não existe sinônimos associados.

5. O sistema cria uma instância de sistema.
6. O sistema emite Msg04 informando que o sinônimo foi cadastrado encerra o Use Case

Diagrama de Colaboração

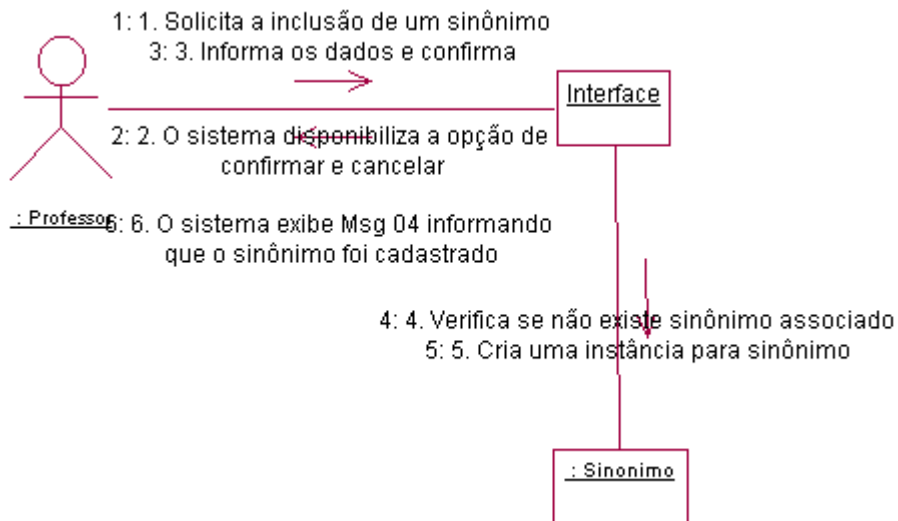


Figura 69 – Diagrama de Colaboração – cadastrarSinonimo – Curso Normal

Curso Alternativo:

Alternativo 1. O professor solicita a busca de um sinônimo.

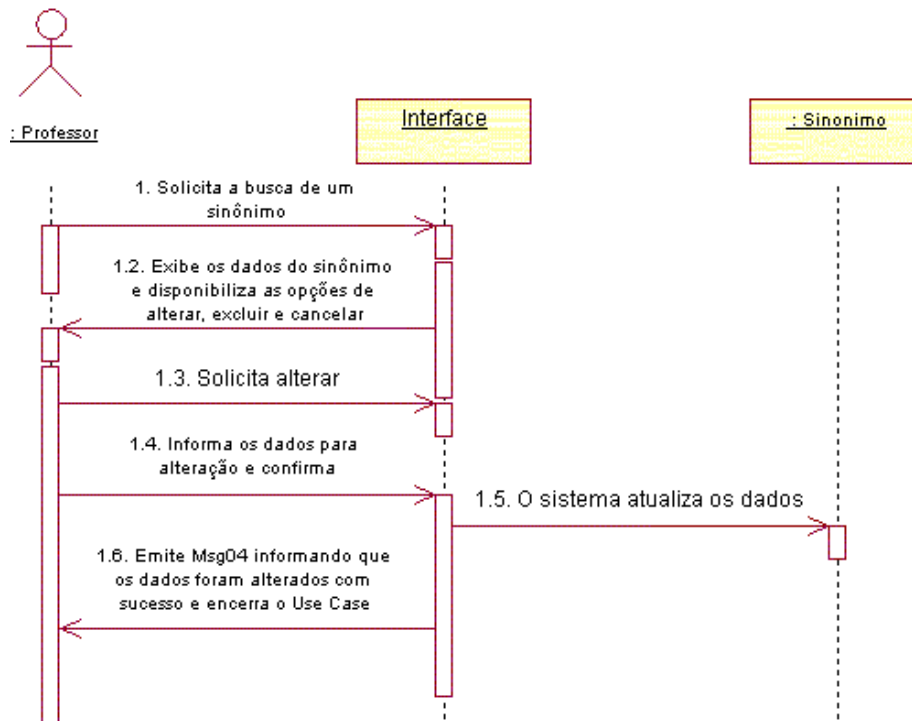


Figura 70 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1

- 1.2. O sistema exibe os dados do sinônimo e disponibiliza as opções de alterar, excluir e cancelar.
- 1.3. O professor solicita alterar.
- 1.4. O professor informa os dados para a alteração do sinônimo e confirma.
- 1.5. O sistema atualiza os dados.
- 1.6. O sistema emite Msg04 informando que o sinônimo foi alterado e encerra o Use Case.

Diagrama de Colaboração

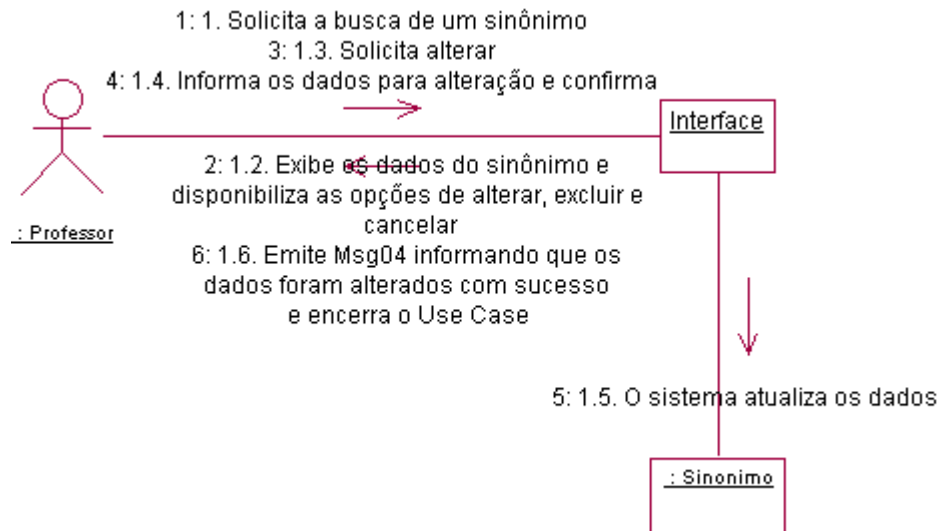


Figura 71 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1

Alternativo 3. O professor solicita cancelar.

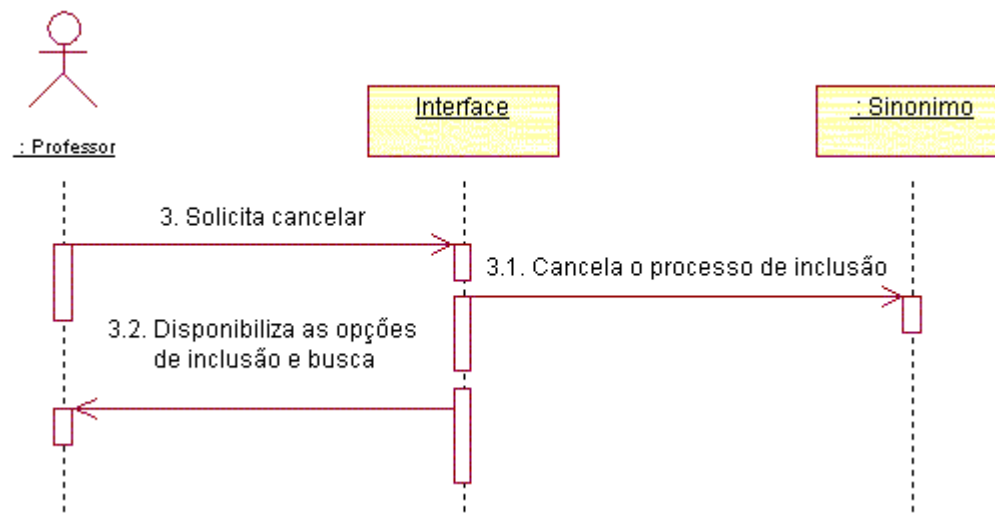


Figura 72 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 3

3.1. O sistema cancela o processo de inclusão.

3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

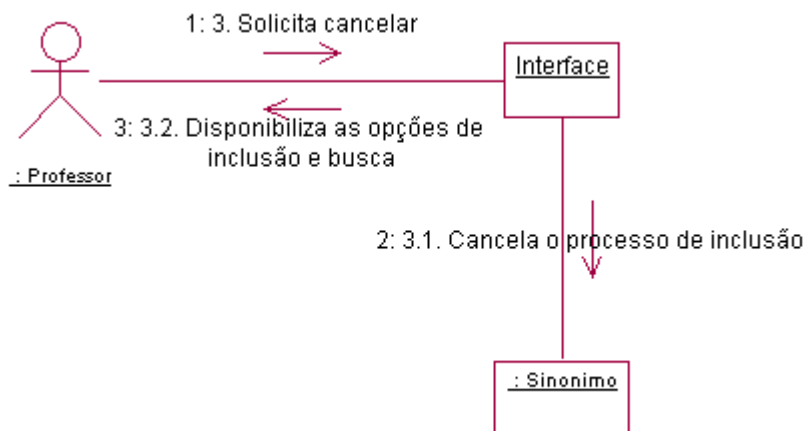


Figura 73 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 3

Alternativo 4. O sistema verifica que existem sinônimos associados.

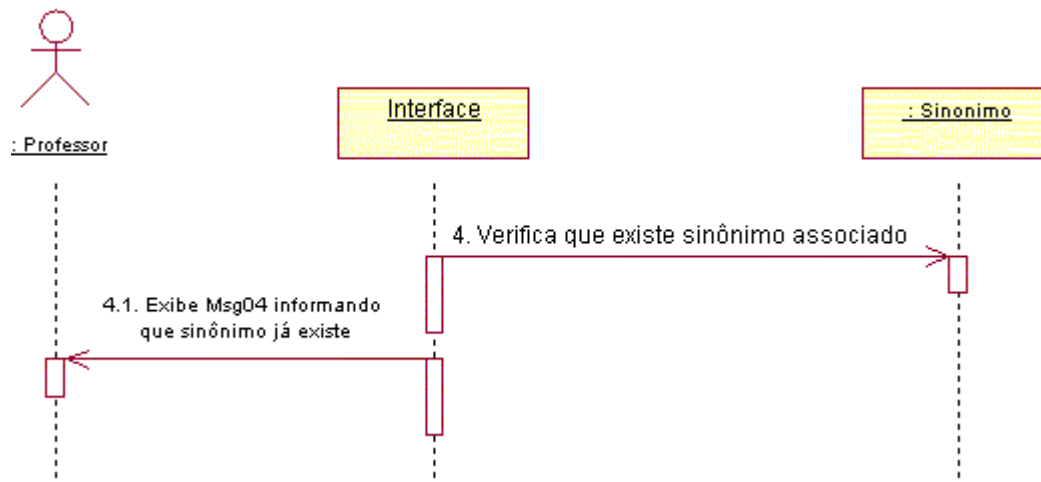


Figura 74 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 4

4.1. O sistema emite Msg04 informando que o sinônimo já existe.

Diagrama de Colaboração

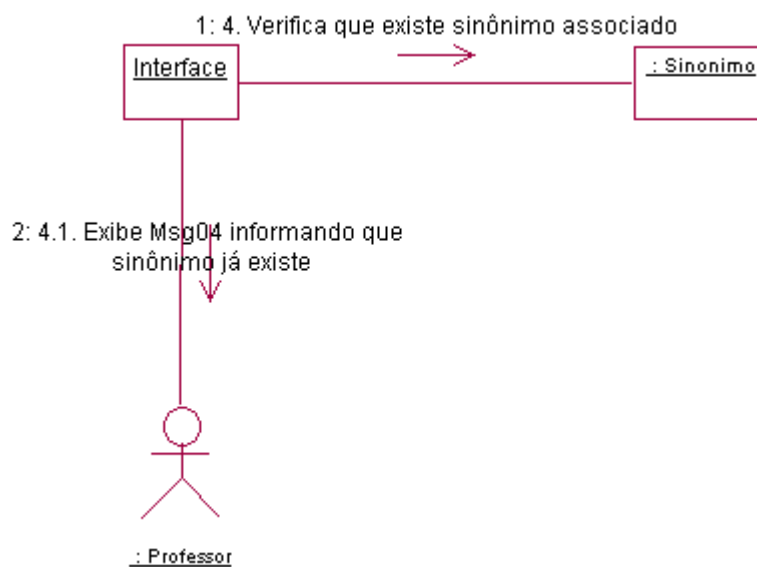


Figura 75 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 4

Alternativo 1.3. O professor escolhe a opção de excluir.

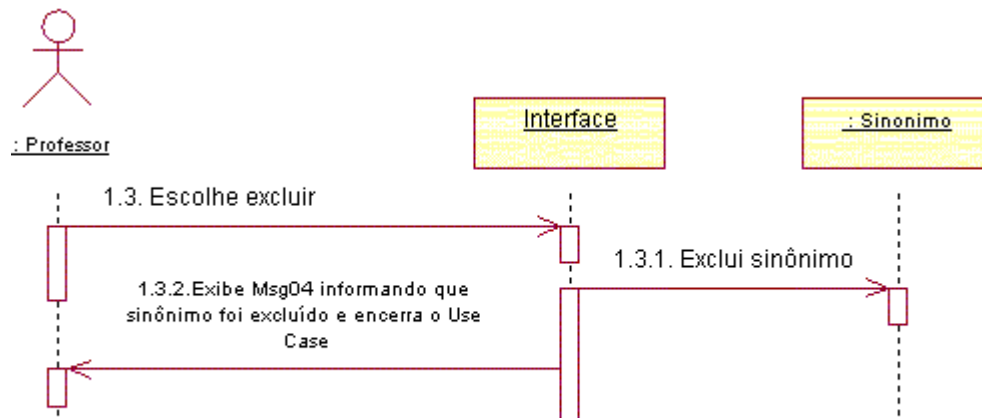


Figura 76 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1.3

1.3.1. O sistema exclui o sinônimo.

1.3.2. O sistema exibe Msg04 informando que o sinônimo foi excluído e encerra o Use Case.

Diagrama de Colaboração

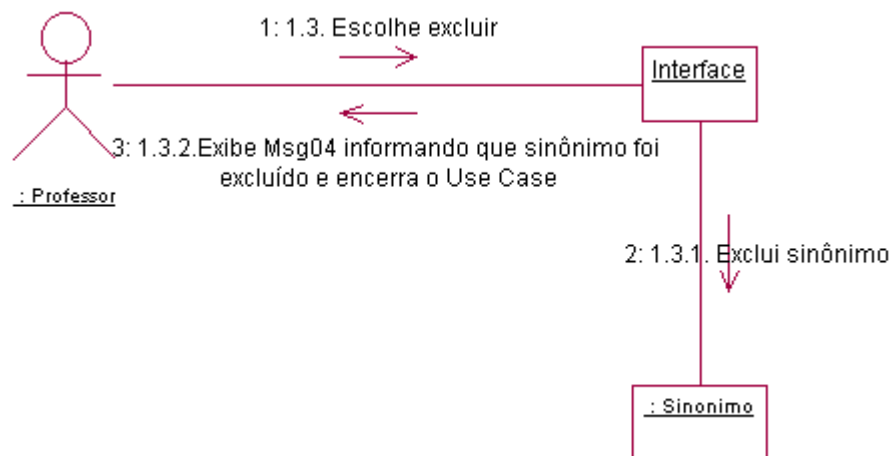


Figura 77 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1.3

Alternativo 1.3. O professor escolhe a opção de cancelar.

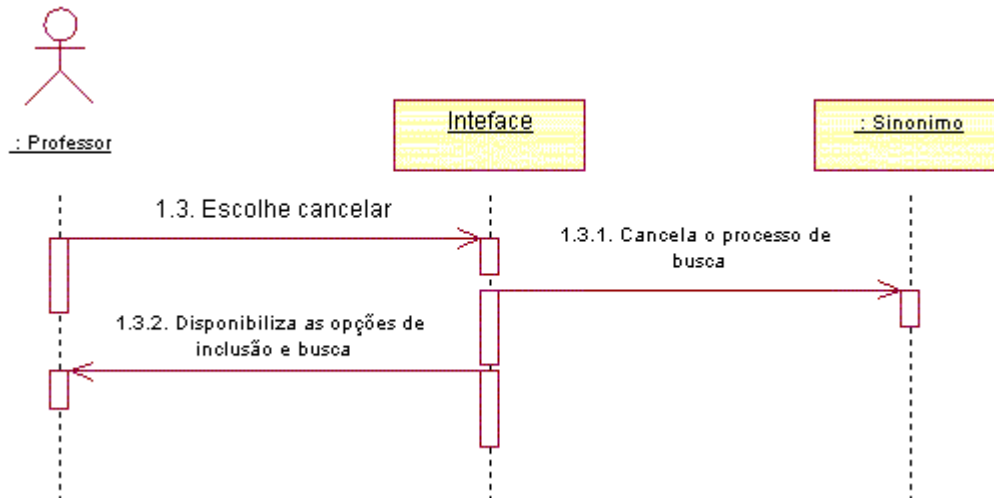


Figura 78 - Use Case 04 – cadastrarSinonimo – Curso Alternativo 1.3

1.3.1. O sistema cancela o processo de busca.

1.3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

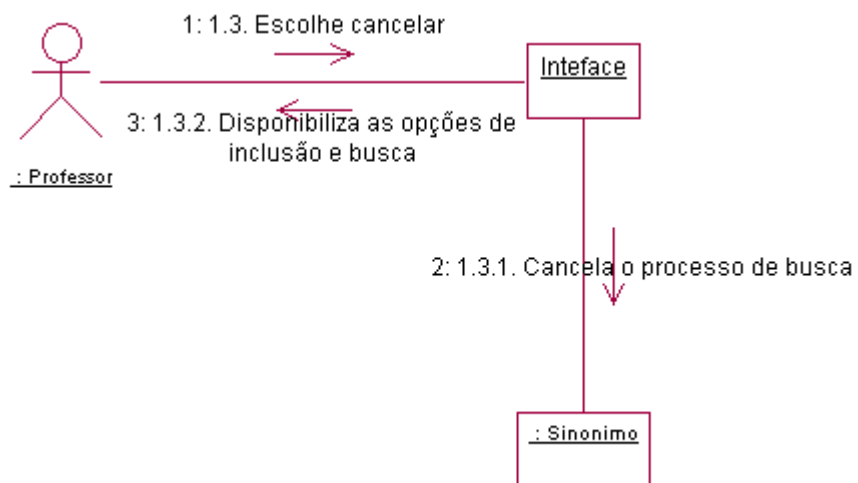


Figura 79 – Diagrama de Colaboração – cadastrarSinonimo – Caso Alternativo 1.3

Número: 05

Use Case: cadastrarPalavraFixa

Descrição: Este Use Case trata do cadastro de palavras fixas.

Ator: Professor

Curso Normal:

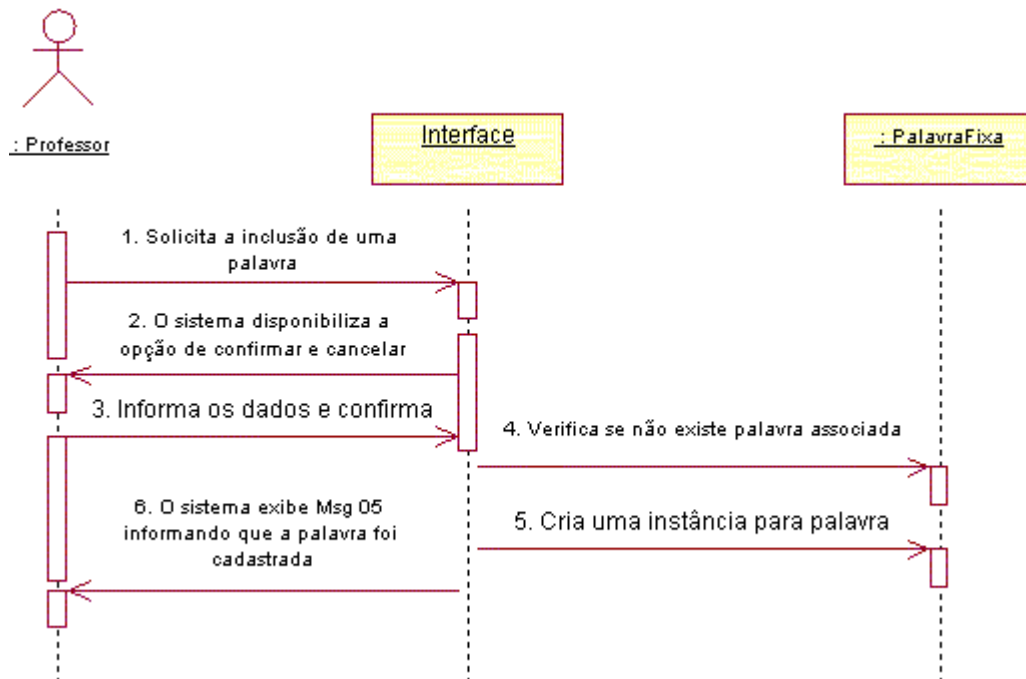


Figura 80 - Use Case 05 – cadastrarPalavraFixa – Curso Normal

1. Professor solicita a inclusão de uma palavra.
2. O sistema disponibiliza a opção de confirmar e cancelar.
3. O professor informa os dados e confirma.
4. O sistema verifica se não existe palavra associada.
5. O sistema cria uma instância para palavra fixa.
6. O sistema emite Msg05 informando que a palavra foi cadastrada e encerra o Use Case

Diagrama de Colaboração

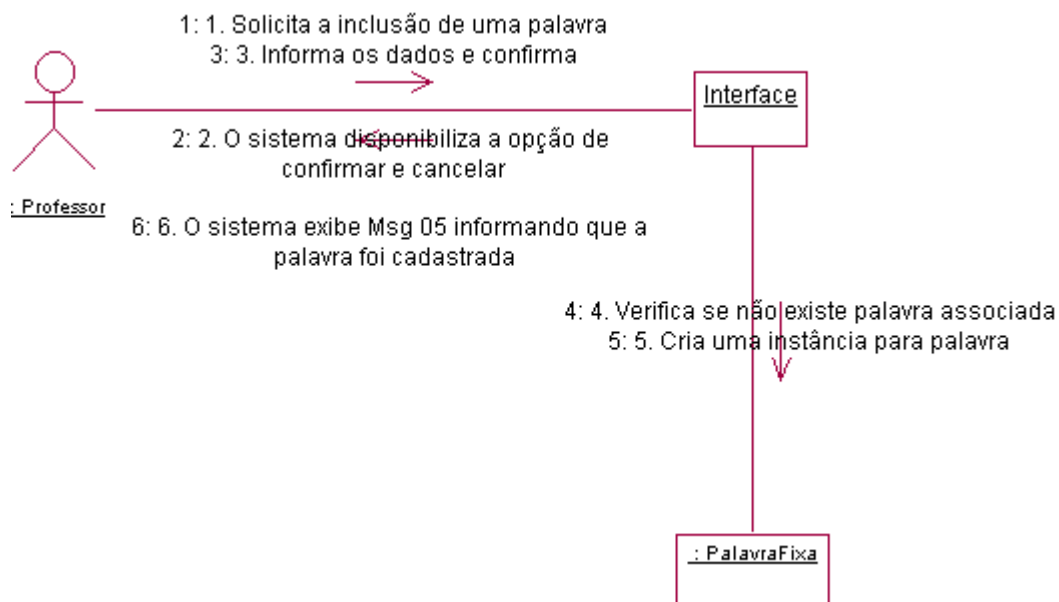


Figura 81 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Normal

Curso Alternativo:

Alternativo 1. O professor solicita a busca de um sinônimo.

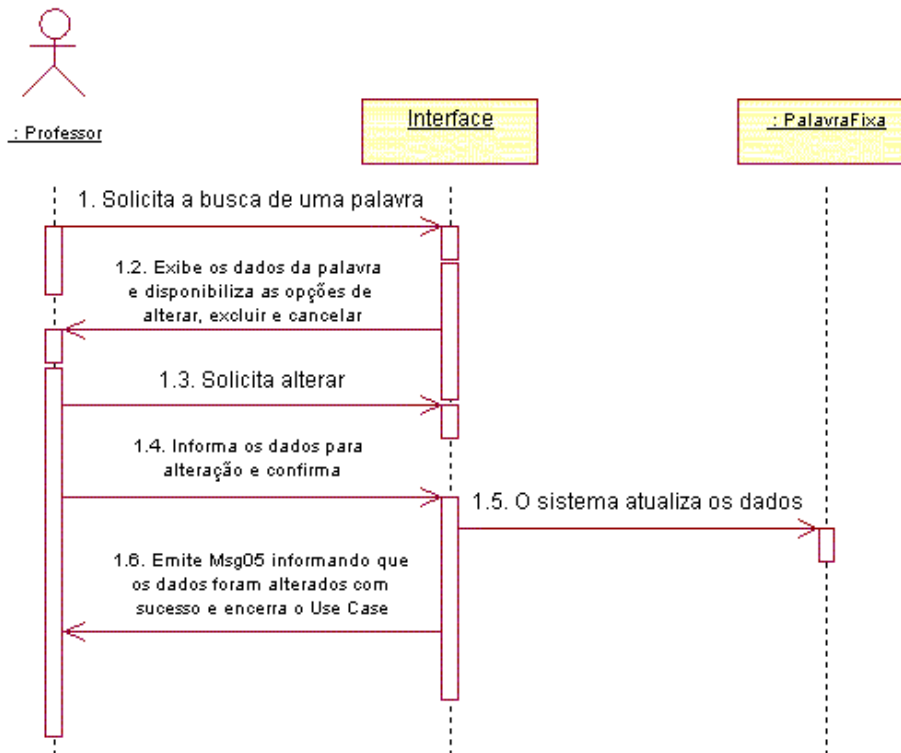


Figura 82 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1

- 1.2. O sistema exibe os dados da palavra e disponibiliza as opções de alterar, excluir e cancelar.
- 1.3. O professor solicita alterar.
- 1.4. O professor informa os dados para a alteração da palavra e confirma.
- 1.5. O sistema atualiza os dados.
- 1.6. O sistema emite Msg05 informando que a palavra foi alterada e encerra o Use Case.

Diagrama de Colaboração

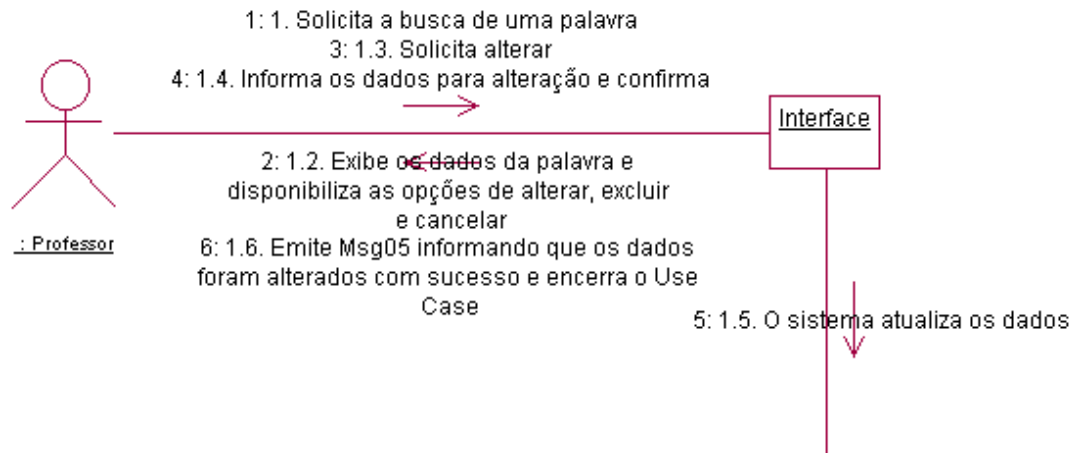


Figura 83 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1

Alternativo 3. O professor solicita cancelar.

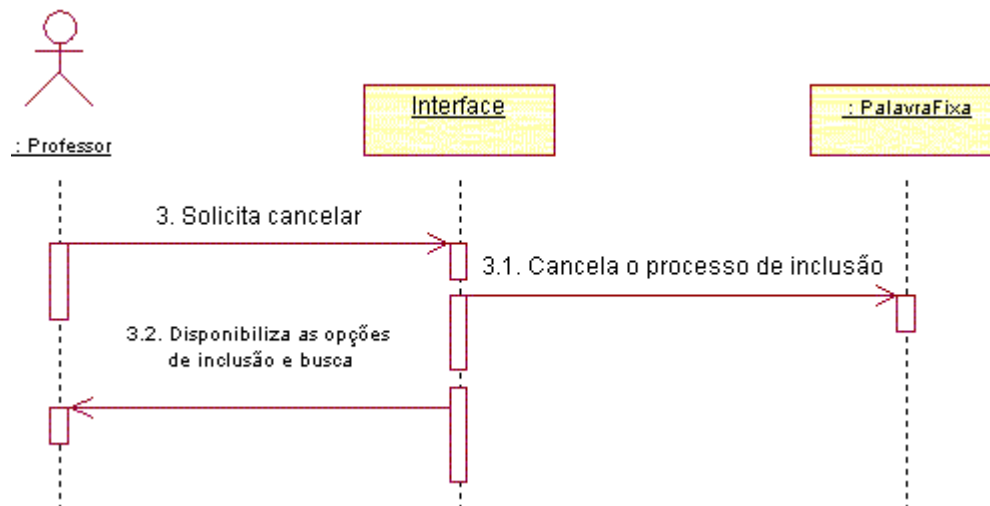


Figura 84 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 3

3.1. O sistema cancela o processo de inclusão.

3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

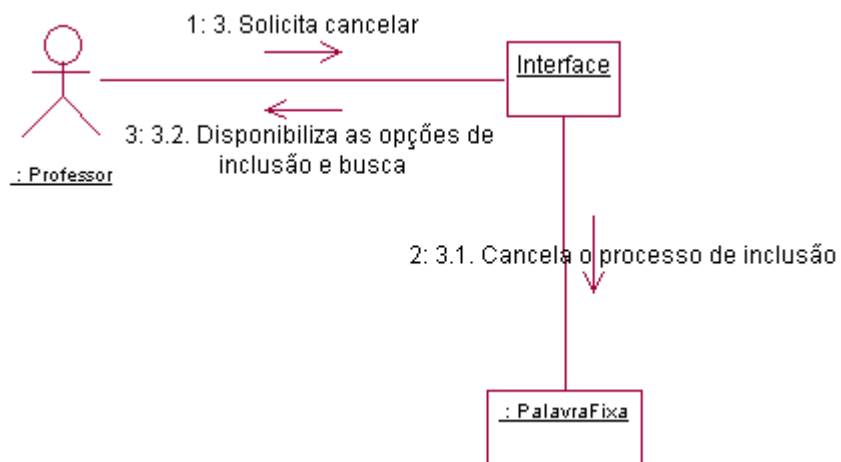


Figura 85 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 3

Alternativo 4. O sistema verifica que existe palavra associada.

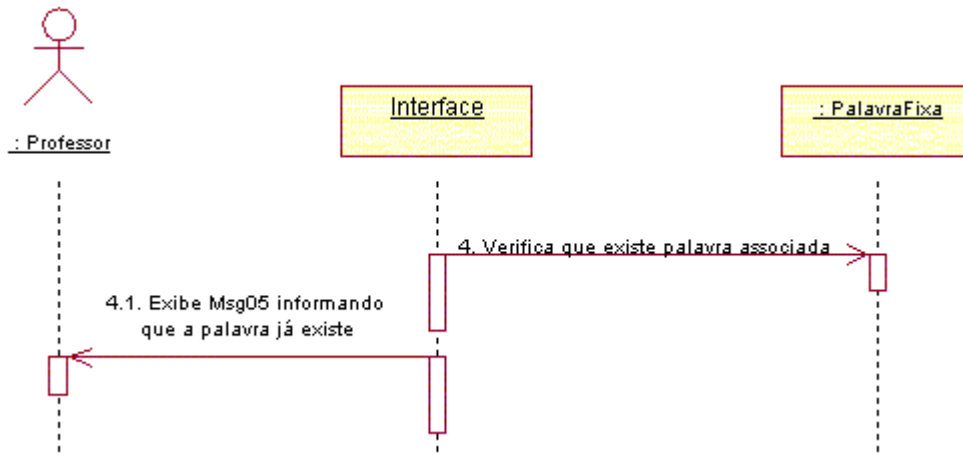


Figura 86 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 4

4.1. O sistema emite Msg05 informando que a palavra já existe.

Diagrama de Colaboração

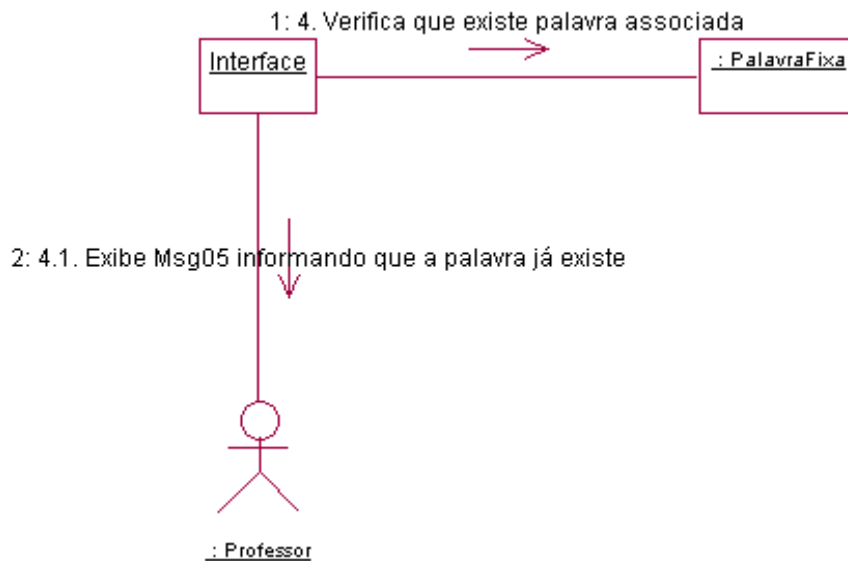


Figura 87 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 4

Alternativo 1.3. O professor escolhe a opção de excluir.

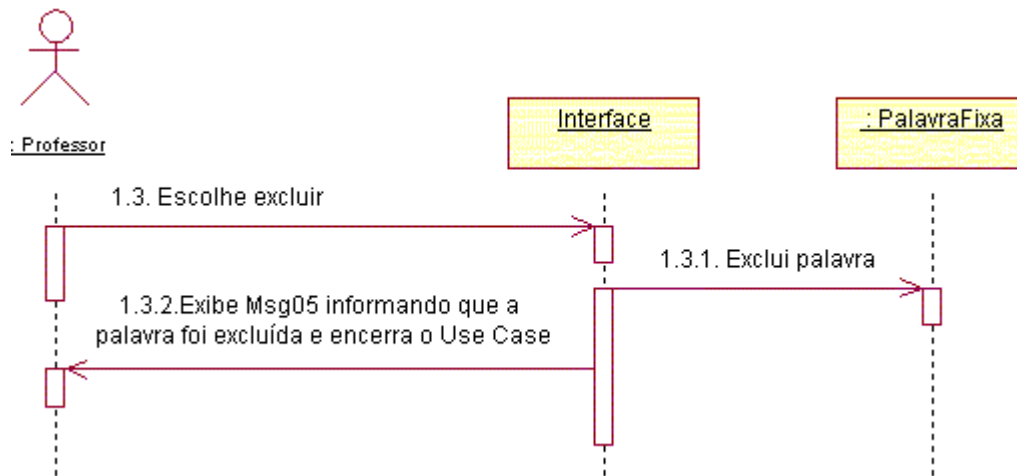


Figura 88 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1.3

1.3.1. O sistema exclui a palavra.

1.3.2. O sistema exibe Msg05 informando que a palavra foi excluída e encerra o Use Case.

Diagrama de Colaboração

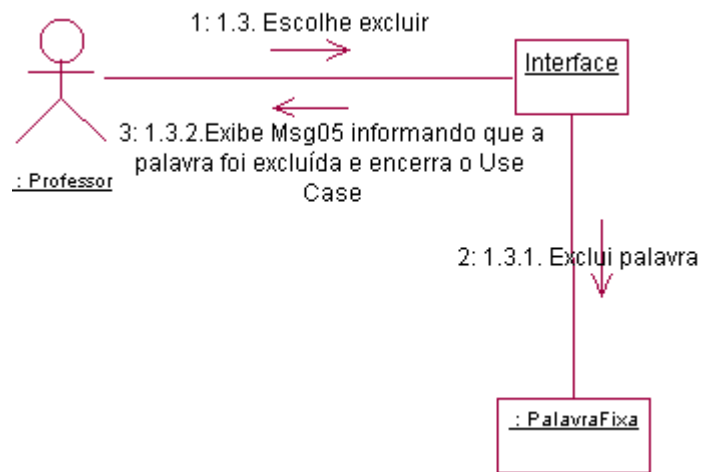


Figura 89 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1.3

Alternativo 1.3. O professor escolhe a opção de cancelar.

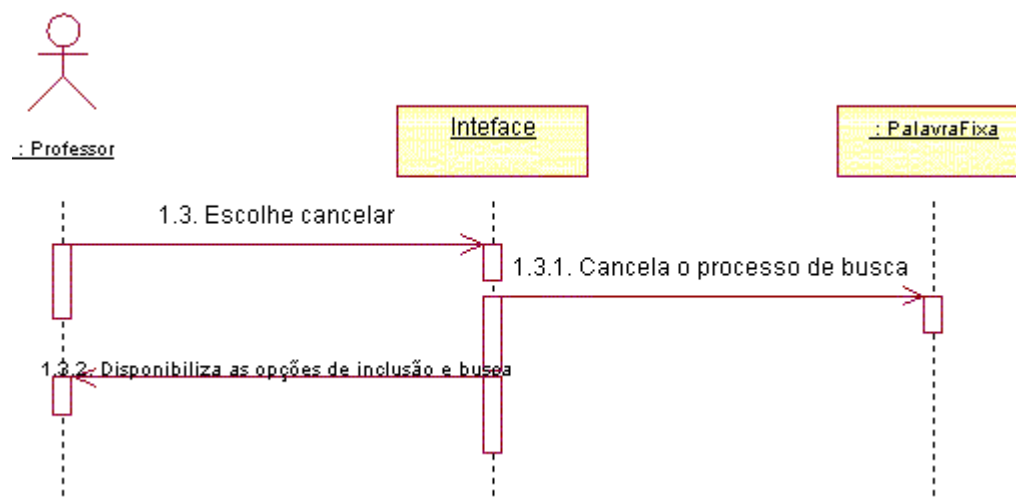


Figura 90 - Use Case 05 – cadastrarPalavraFixa – Curso Alternativo 1.3

1.3.1. O sistema cancela o processo de busca.

1.3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

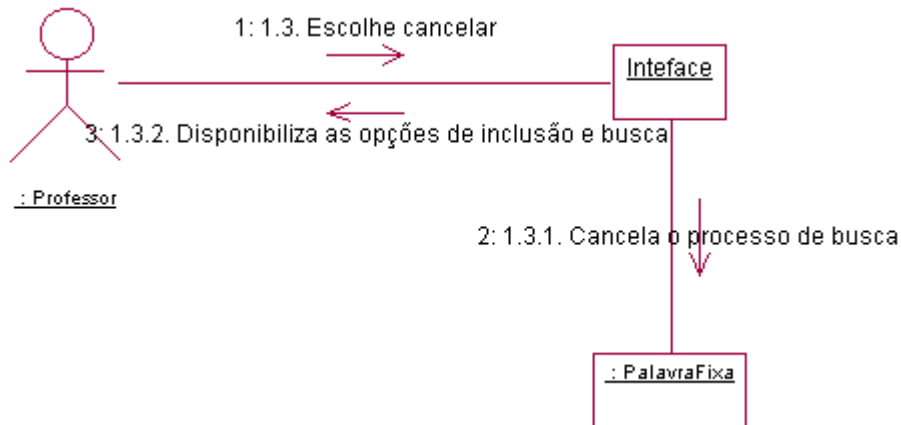


Figura 91 – Diagrama de Colaboração – cadastrarPalavraFixa – Curso Alternativo 1.3

Número: 06

Use Case: cadastrarPalavraNaoAtomica

Descrição: Este Use Case trata do cadastro de palavras não atômicas.

Ator: Professor

Curso Normal:

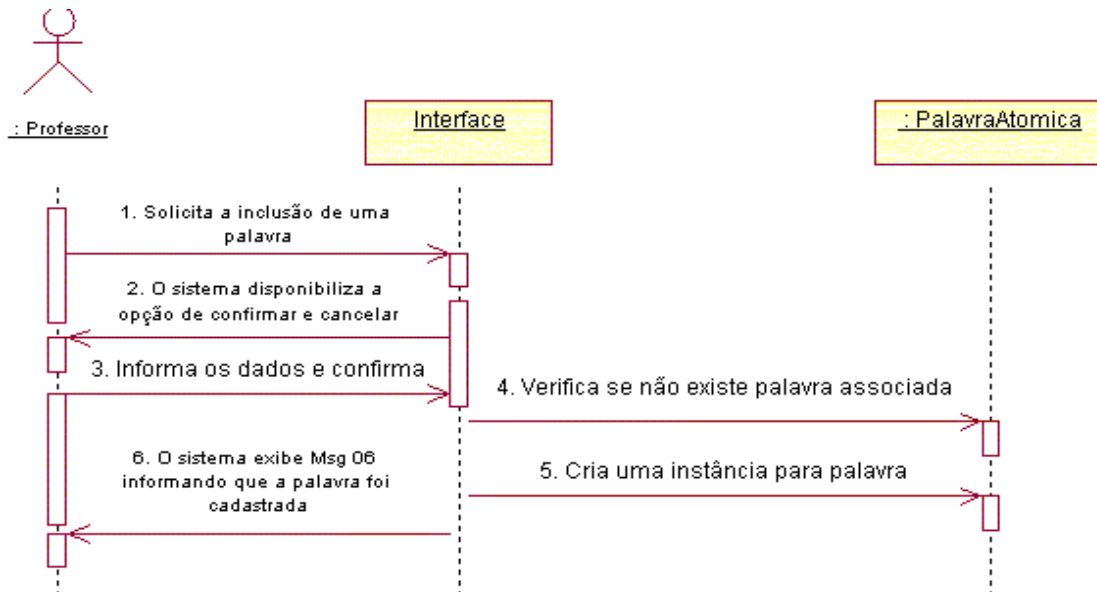


Figura 92 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Normal

1. Professor solicita a inclusão de uma palavra não atômica.
2. O sistema disponibiliza a opção de confirmar e cancelar.
3. O professor informa os dados e confirma.
4. O sistema verifica se não existe palavra associada.
5. O sistema cria uma instância de palavra não atômica.
6. O sistema emite Msg06 informando que a palavra foi cadastrada e encerra o Use Case

Diagrama de Colaboração

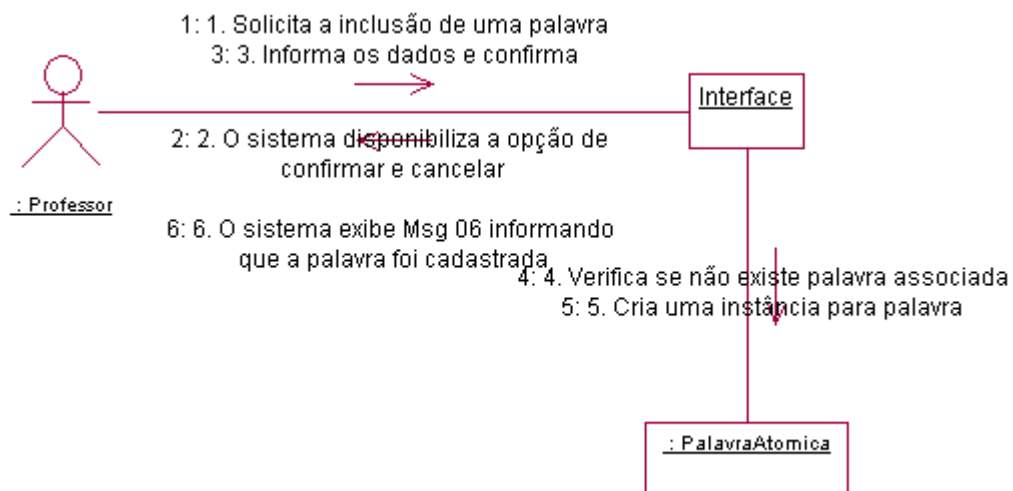


Figura 93 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Normal

Curso Alternativo:

Alternativo 1. O professor solicita a busca de uma palavra.

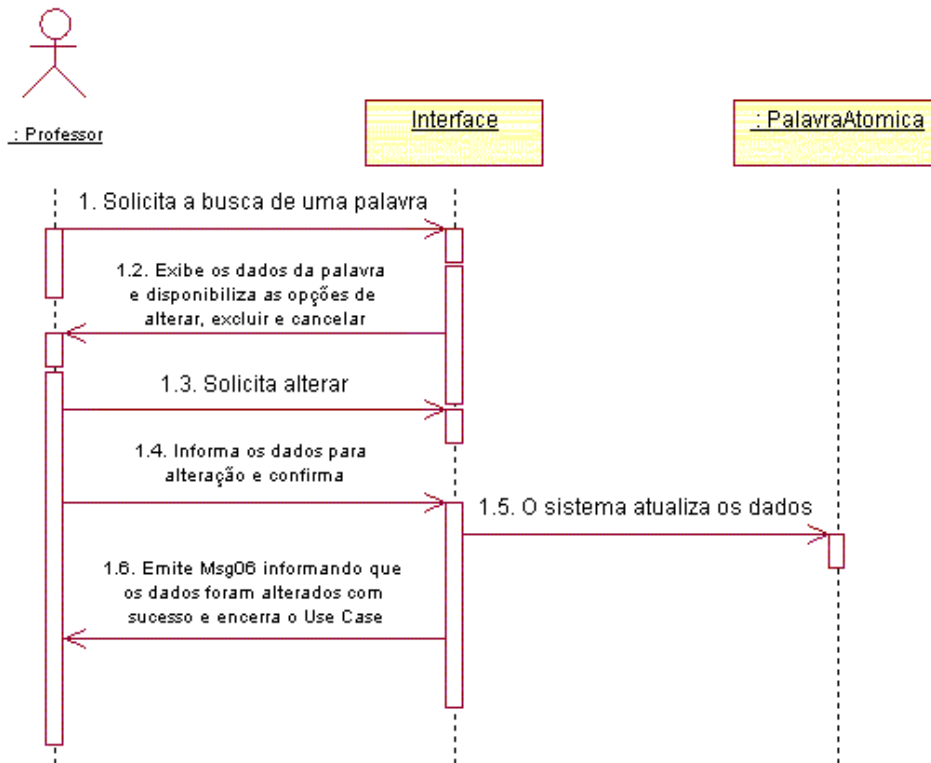


Figura 94 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 1

- 1.2. O sistema exibe a palavra e disponibiliza as opções de alterar, excluir e cancelar.
- 1.3. O professor solicita alterar.
- 1.4. O professor informa os dados para a alteração da palavra e confirma.
- 1.5. O sistema emite Msg06 informando que a palavra foi alterada e encerra o Use Case.

Diagrama de Colaboração.

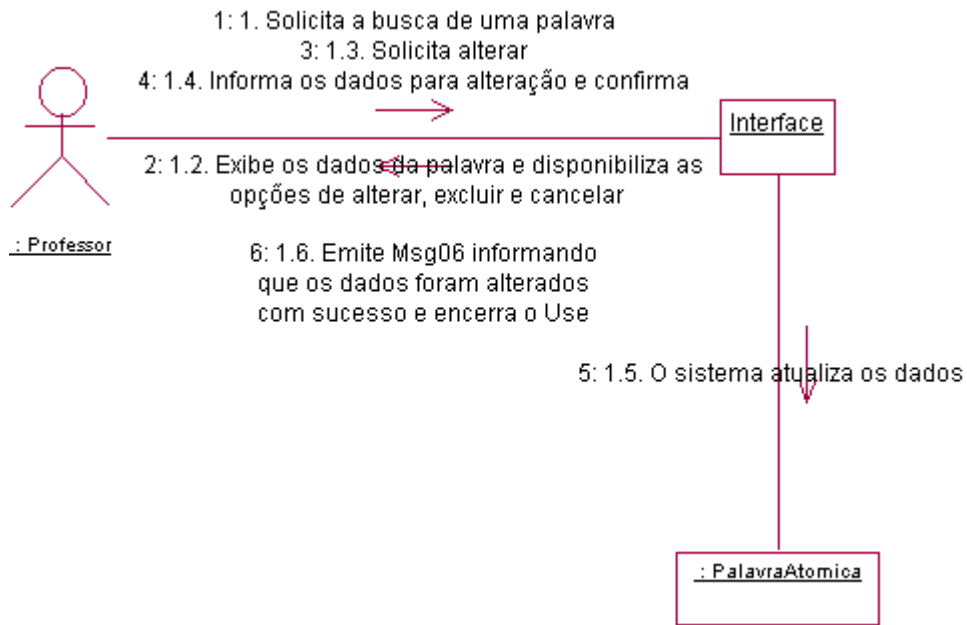


Figura 95 – Diagrama de Colaboração– cadastrarPalavraNaoAtomica – Curso Alternativo 1

Alternativo 3. O professor solicita cancelar.

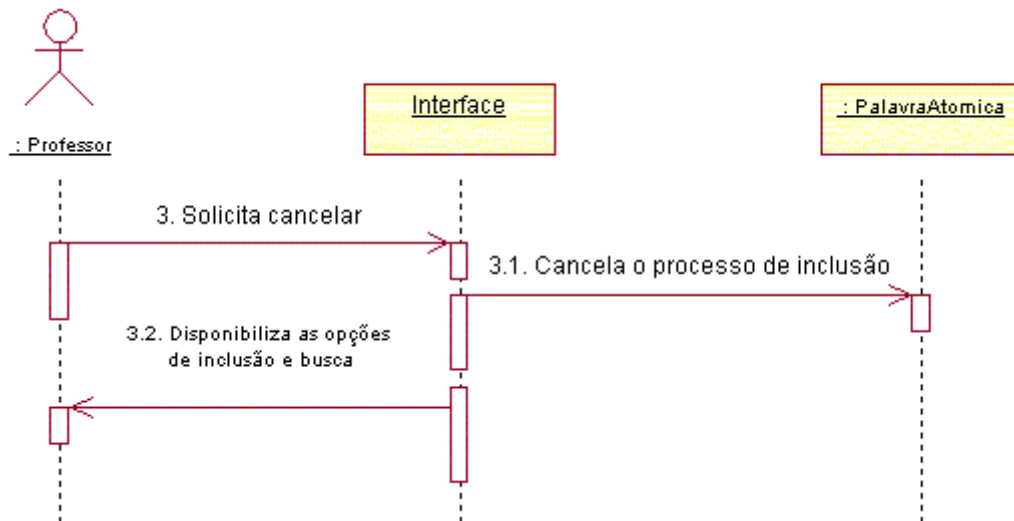


Figura 96 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 3

3.1. O sistema cancela o processo de inclusão.

3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

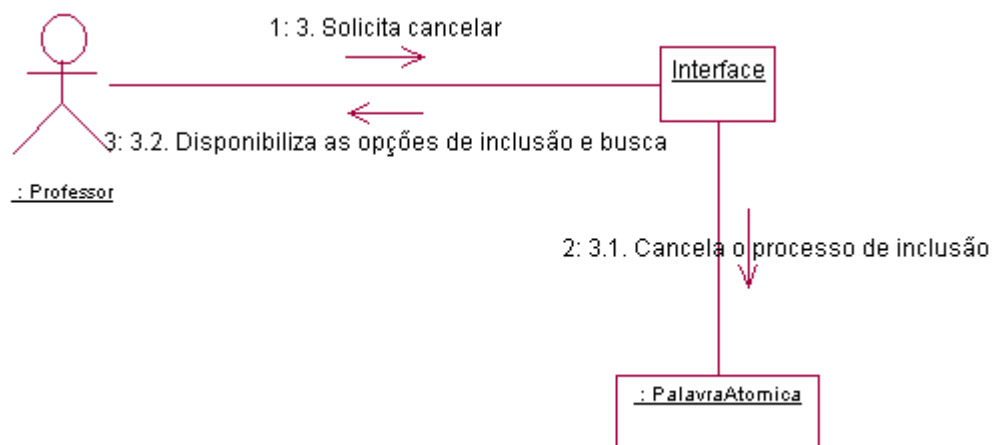


Figura 97 - Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 3

Alternativo 4. O sistema verifica que existe palavra associada.

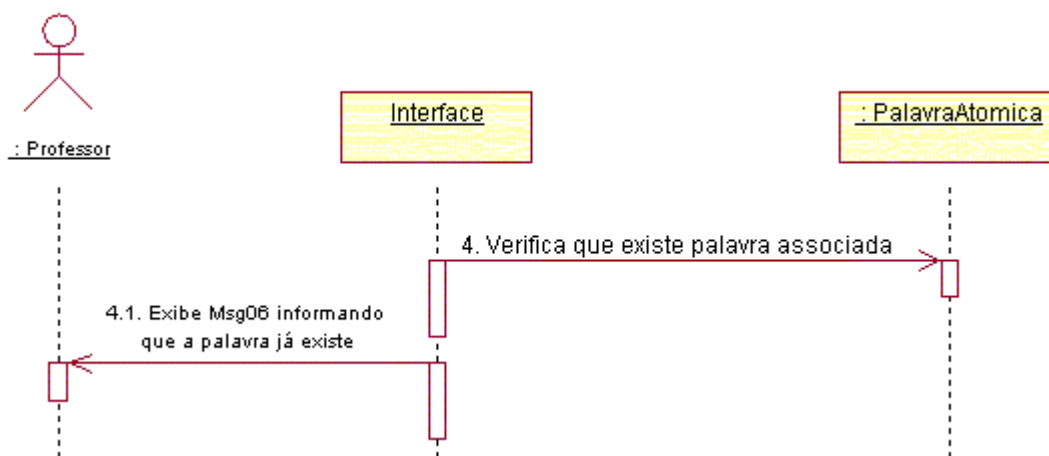


Figura 98 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 4

4.1. O sistema emite Msg06 informando que a palavra já existe.

Diagrama de Colaboração

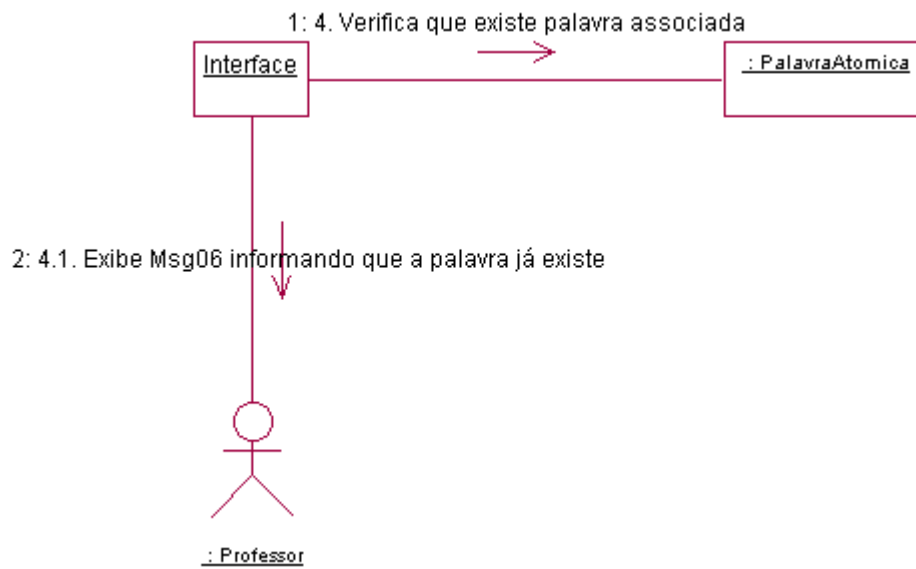


Figura 99 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 4

Alternativo 1.3. O professor escolhe a opção de excluir.

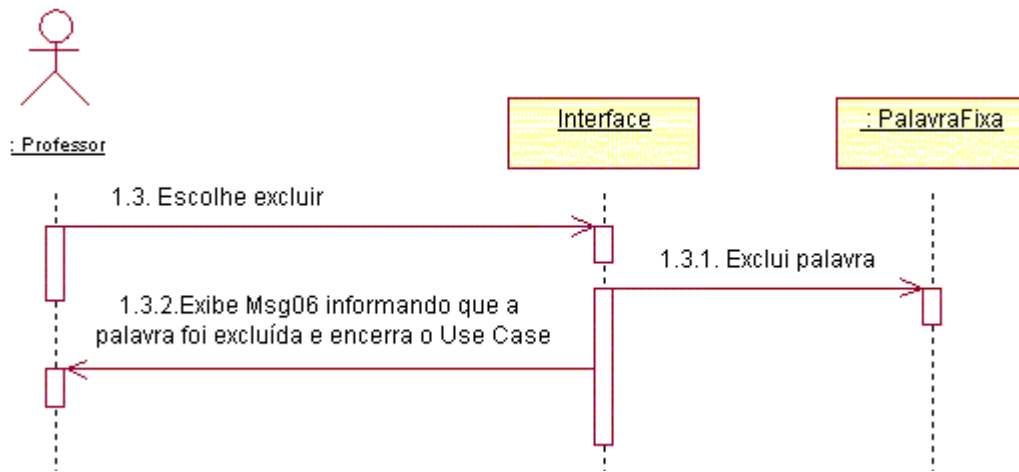


Figura 100 - Use Case 06 – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3

1.3.1. O sistema exclui a palavra.

1.3.2. O sistema exibe Msg06 informando que a palavra foi excluída e encerra o Use Case.

Diagrama de Colaboração

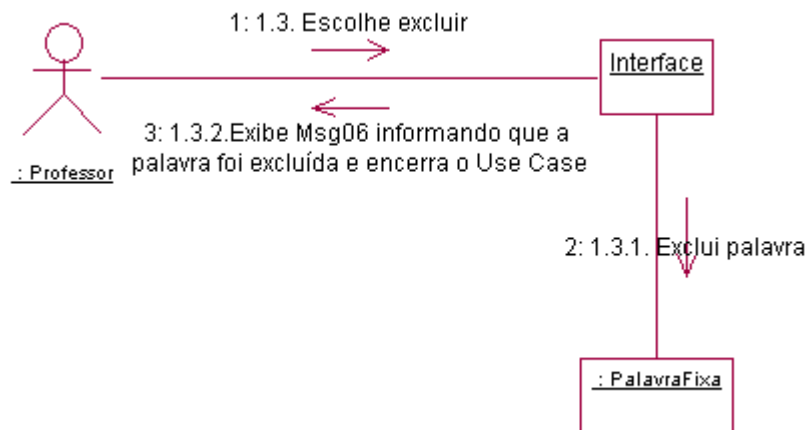


Figura 101 – Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3

Alternativo 1.3. O professor escolhe a opção de cancelar.

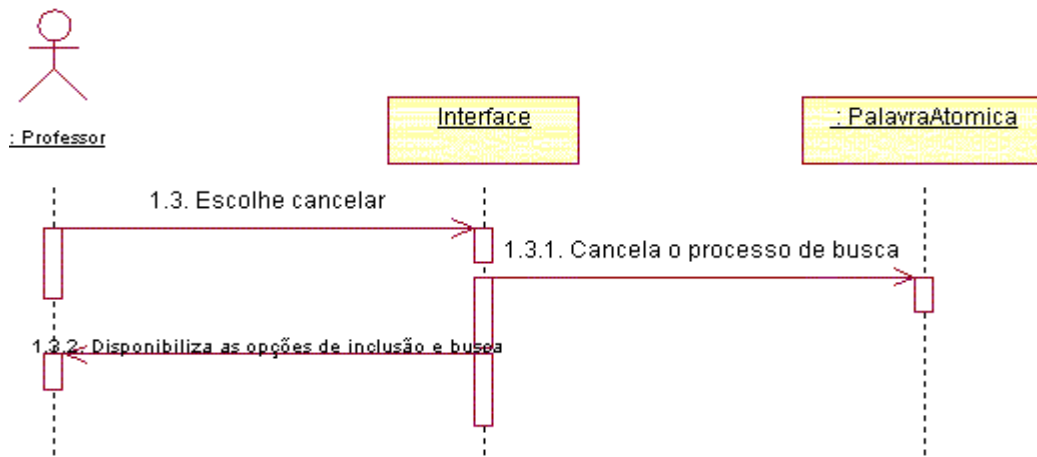


Figura 102 - Use Case 06 – cadastrarPalavraNaoAtomica – Caso Alternativo 1.3

1.3.1. O sistema cancela o processo de busca.

1.3.2. O sistema disponibiliza as opções de inclusão e busca.

Diagrama de Colaboração

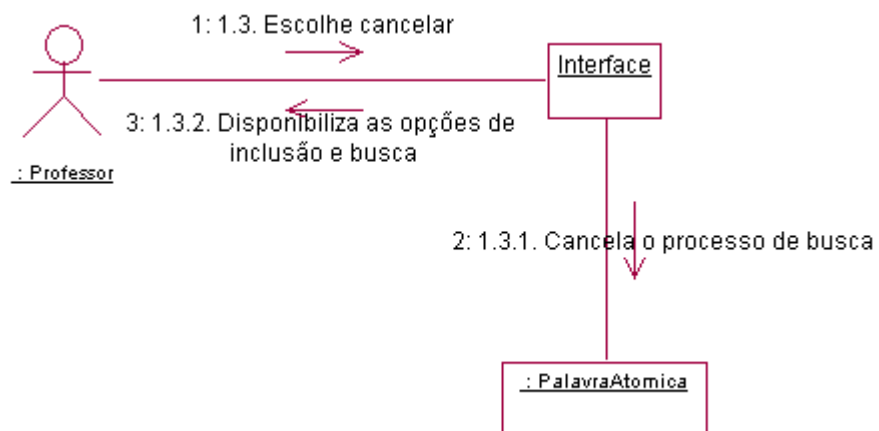


Figura 103 - Diagrama de Colaboração – cadastrarPalavraNaoAtomica – Curso Alternativo 1.3

5.2.4 Diagrama de classe

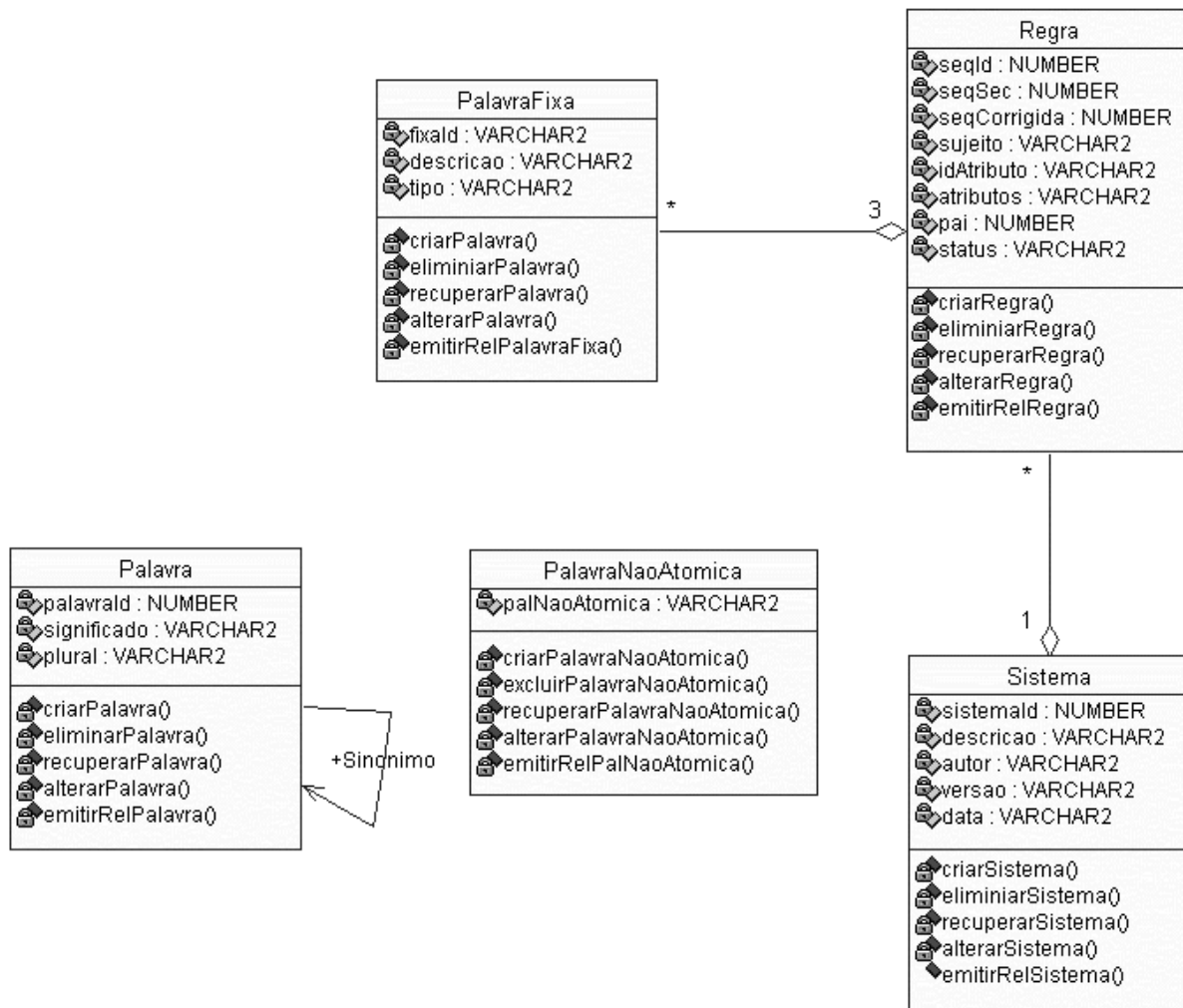


Figura 104 - Diagrama de Classes de Análise

5.3 Documentação de Projeto

5.3.1 Descrição das decisões de projeto

Hardware: Para a utilização do software, o usuário deverá possuir, no mínimo, um computador Pentium II ou similar com 400Mhz, 64 Mb de memória Ram e uma impressora.

Software: Desenvolvido utilizando-se da linguagem de programação Java, utilizando como sistema operacional Windows 98 e base de dados Access e Oracle.

Inicialmente o software deverá ser instalado em um computador mono-usuário, porém, posteriormente o sistema será migrado para trabalhar em rede suportado por servidor de Internet Apache e utilizado por clientes através de navegadores de Internet padrão de mercado.

5.3.2 Refinamento dos diagramas na fase de análise

5.3.2.1 Diagramas de use cases

5.3.2.1.1 Lista de use case

1. cadastrarSistema
2. cadastrarPalavra
3. cadastrarRegra
4. cadastrarSinonimo
5. cadastrarPalFixa
6. cadastrarPalNaoAtomica
7. gerarRelPalFixa
8. gerarRelPalNaoAtomica
9. gerarRelatorioSistema
10. gerarRelRegraOriginal
11. gerarRelatorioPalavra
12. gerarRelatorioRegra

5.3.2.1.2 Lista eventos

Nº	Use Case	Descrição	Evento	Resposta
01	cadastrarSistema	Aluno solicita cadastrar sistema	dadosSistema	Msg01
02	cadastrarPalavra	Aluno solicita cadastrar palavra	dadosPalavra	Msg02
03	cadastrarRegra	Aluno solicita cadastrar regra	dadosRegra	Msg03
04	cadastrarSinonimo	Professor solicita cadastrar sinônimo	dadosSinonimo	Msg04
05	cadastrarPalFixa	Professor solicita cadastrar palavra fixa	dadosPalavraFixa	Msg05
06	cadastrarPalNaoAtomica	Professor solicita cadastrar palavra não atômica	dadosPalavraNaoAtomica	Msg06
07	gerarRelPalFixa	Professor solicita relatório de palavra fixa	solicitaRelatorio	relPalFixa
08	gerarRelPalNaoAtomica	Professor solicita relatório de palavra não atômica	solicitaRelatorio	relPalNaoAtomica
09	gerarRelatorioSistema	Professor solicita relatório de sistema	solicitaRelatorio	relSistema
10	gerarRelRegraOriginal	Professor solicita relatório de regra original	solicitaRelatorio	relRegraOriginal
11	gerarRelatorioPalavra	Professor solicita relatório de palavra	solicitaRelatorio	relPalavra
12	gerarRelatorioRegra	Professor solicita relatório de regra	solicitaRelatorio	relRegra

Observações:

Msg01 = Sistema Cadastrado / Sistema Alterado / Sistema Excluído / Sistema já existe / Não será possível excluir o sistema existem regras associadas a ele

Msg02 = Palavra Cadastrada / Palavra Alterada / Palavra Excluída / Palavra já existe

Msg03 = Regra Cadastrada / Regra Alterada com sucesso / Regra Excluída / Existe erro na frase, deseja continuar?

Msg04 = Sinônimo Cadastrado / Sinônimo Alterado com sucesso / Sinônimo Excluído

Msg05 = Palavra Cadastrada / Palavra Alterada com sucesso / Palavra Excluída / Palavra já existe

Msg06 = Palavra Cadastrada / Palavra Alterada com sucesso / Palavra Excluída / Palavra já existe

5.3.2.1.3 Diagrama de use case

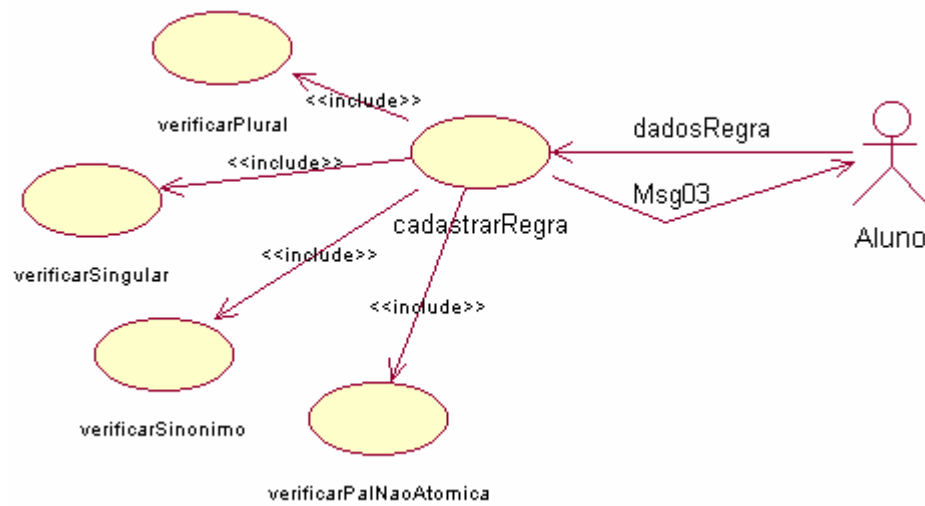


Figura 105 - Diagrama de Use Case – Cadastrar Regras

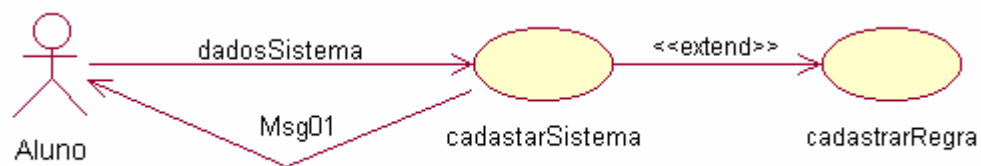


Figura 106 - Diagrama de Use Case – Cadastrar Sistema

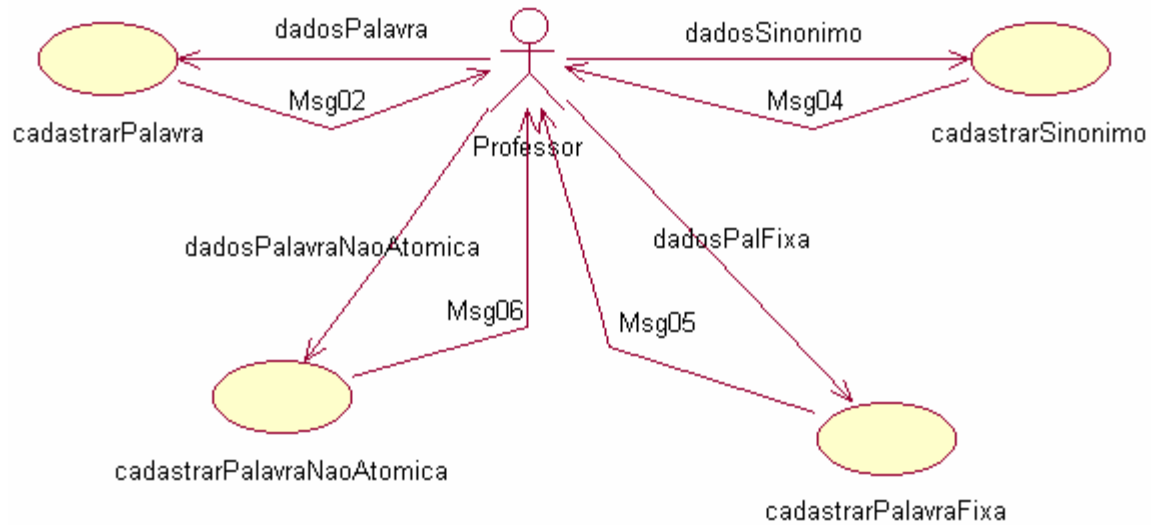


Figura 107 - Diagrama de Use Case – Cadastros da Base de Apoio

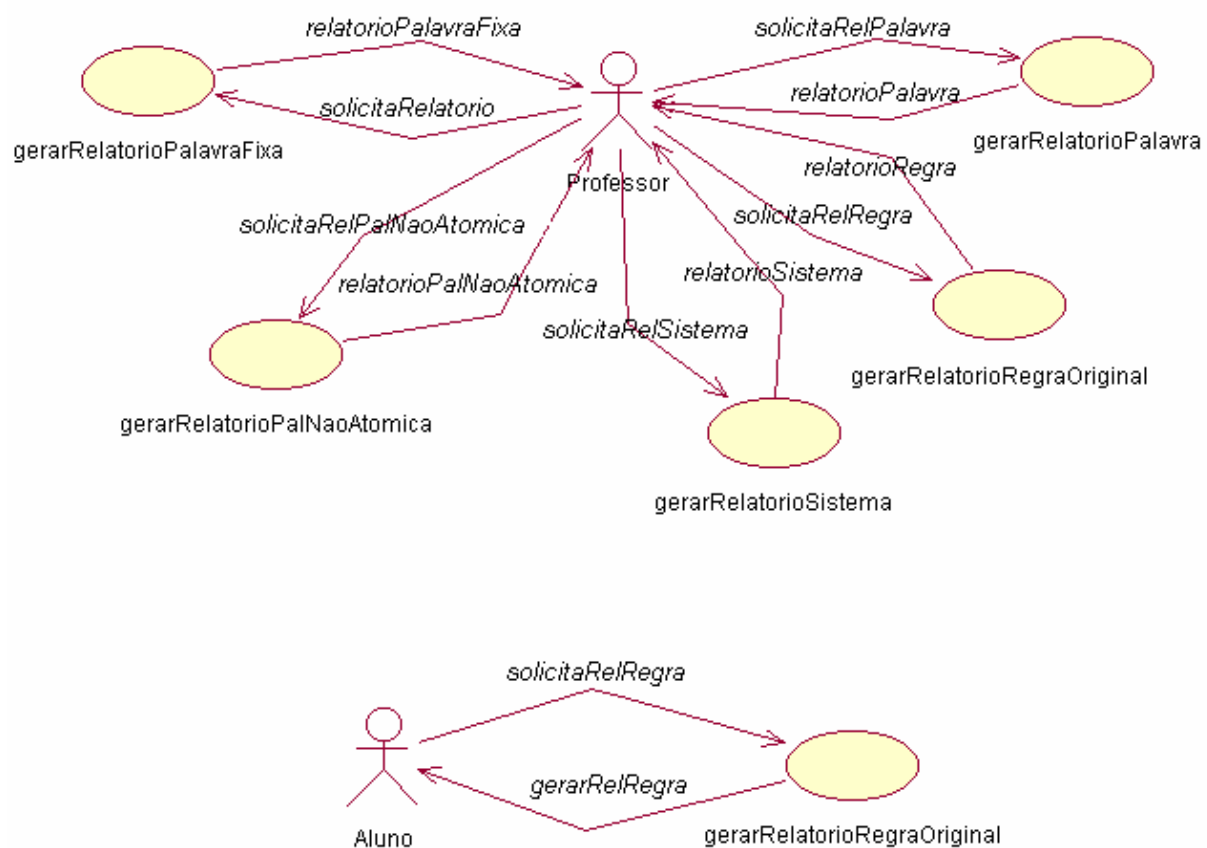


Figura 108 - Diagrama de Use Case - Relatórios

5.3.2.2 Diagramas de estrutura estática

5.3.2.2.1 Diagrama de classe

5.3.3 Mapeamento do diagrama de classe para o banco de dados relacional

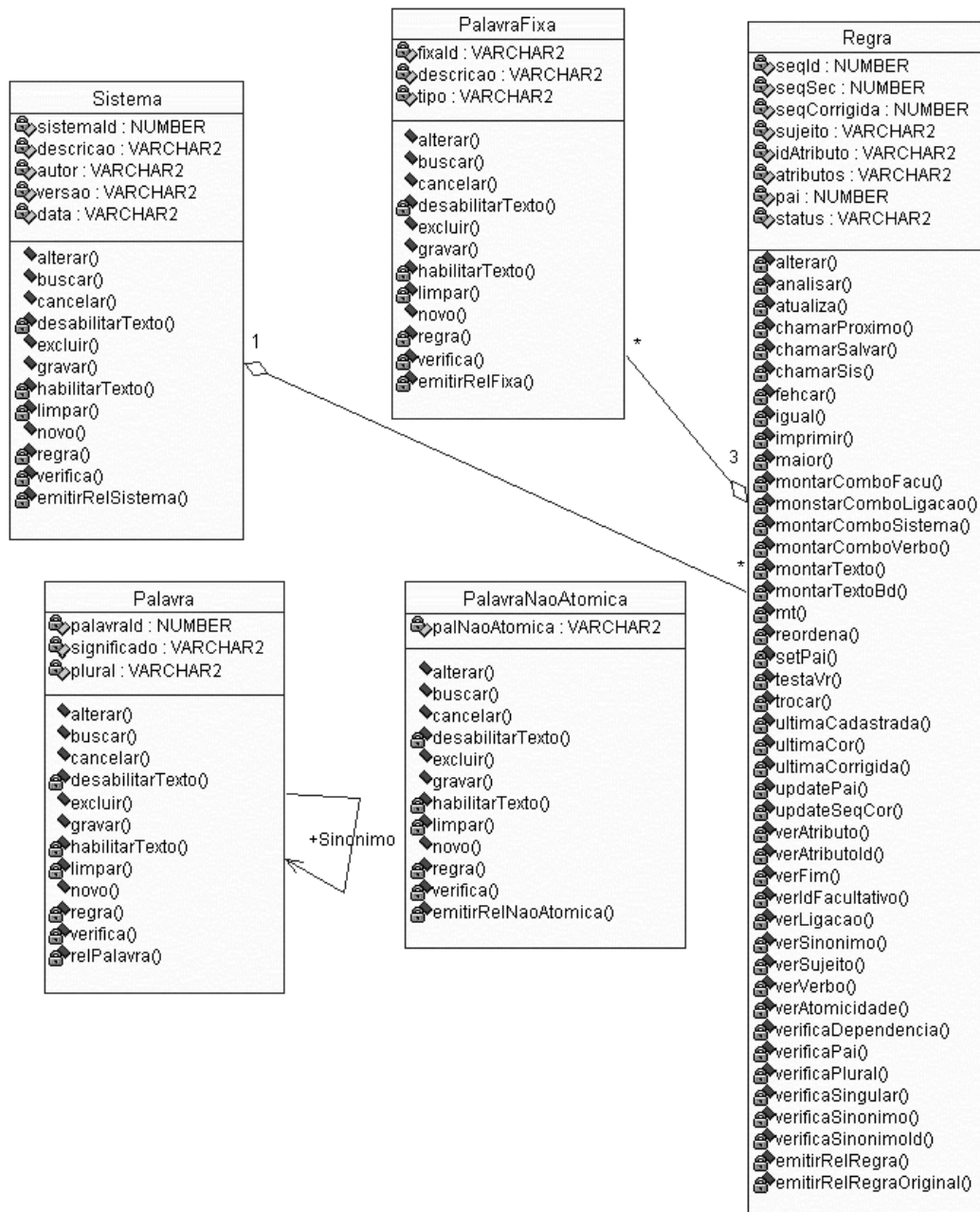


Figura 109 – Mapeamento do Diagrama de Classes

Palavra	(<u>palavraId</u> , significado, plural)
Sinonimo	(<u>palavraId</u> , <u>sinonimoId</u>)
Sistema	(<u>sistemaId</u> , descricao, versao, autor, data)
PalavraFixa	(<u>fixaId</u> , descricao, tipo)
Frase	(<u>sistemaId</u> , <u>seqId</u> , <u>seqSec</u> , seqCor, <u>fixaId</u> , sujeito, <u>fixaId</u> , idAtributo, <u>fixaId</u> , atributos, status, pai)

5.3.4 Descrição da padronização do sistema

- As iniciais do menu estão com a inicial em maiúsculo, possuem fonte Arial 10 e cor preta;
- A seleção do menu possui a cor azul;
- Os itens do menu são agrupados por tipo;
- Todas as telas estão centralizadas;
- Os títulos das telas estão de acordo com a função;
- Todas as telas possuem a cor cinza;
- Os campos e rótulos possuem alinhamento a esquerda;
- Os rótulos de cada campo possuem fonte Arial, tamanho 10 e cor azul;
- Os botões ficam na parte superior com a legenda de suas funções;
- As legendas dos botões estão com a inicial em maiúscula e em preto
- Após a mensagem de erro o cursor volta no campo de erro;
- As mensagens de erro são padronizadas a cada tela por possuírem mensagens específicas

5.3.5 Prototipação da interface

5.3.5.1 Menu Principal

Tela inicial de entrada do sistema onde o usuário poderá informar qual processo deseja realizar.

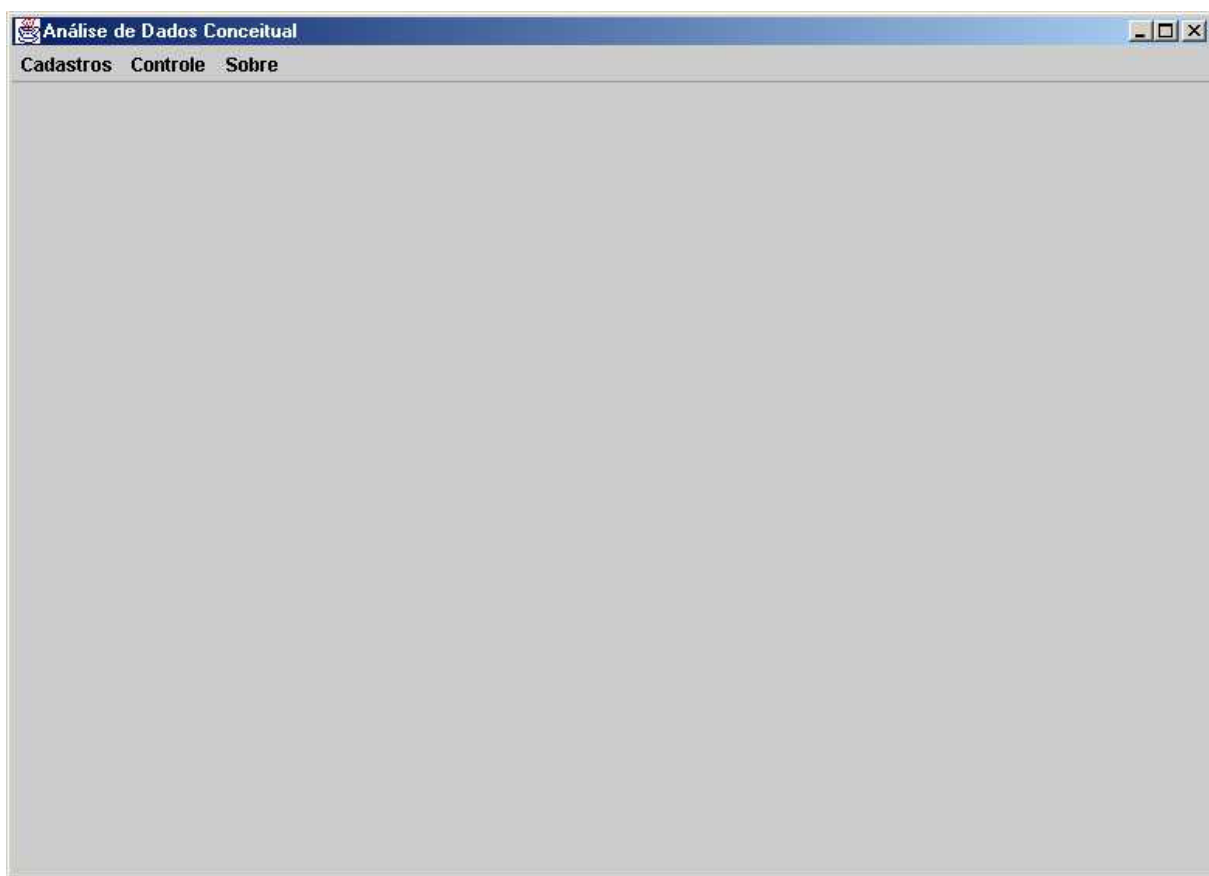


Figura 110 - Menu Principal do Sistema

5.3.5.2 - Cadastrar Sistema

A tela de cadastro de sistema será utilizada para identificar qual é o sistema que será informado, indicando seu nome, breve descrição introdutória sobre o sistema e nome do autor. Nesta tela também será possível acompanhar a versão do sistema e a data da última alteração. A versão do sistema será composta por 3 níveis e será atualizada automaticamente pelo sistema, sendo que o primeiro nível será o nível de conceitos do sistema, ou seja, o sistema receberá o número 1 no início das atividades e será incrementado de 1 sempre que o autor atualizar a versão do cadastro geral do sistema. O 2º nível será atualizado a cada vez que o usuário atualizar as regras do sistema e o 3º nível será atualizado a cada reordenação das regras do sistema de informação.



A imagem mostra uma janela de software intitulada "Cadastro do Sistema". No topo, há uma barra de título azul com o ícone de uma pasta e o texto "Cadastro do Sistema". Abaixo, o título "CADASTRO DO SISTEMA" aparece em uma fonte menor. O formulário contém campos rotulados "Nome:", "Descrição:", "Autor:", "Versão:" e "Data:". O campo "Descrição:" é um texto de área maior. Na base da janela, há duas linhas de botões. A primeira linha contém "Novo", "Alterar", "Excluir", "Buscar", "Gravar" e "Cancelar". A segunda linha contém "<<", "<", ">", ">>", "Regras" e "Sair".

Figura 111 – Tela de cadastro do sistema

5.3.5.3 - Cadastrar Regra

Após a identificação do sistema o usuário deverá indicar as regras do sistema, informando cada uma dentro das normas estabelecidas pelo passo 1 da técnica de Lista-Diagrama, ou melhor, as regras deverão ser informadas respeitando as normas de Lista de Dependências. Nesta tela ainda o usuário efetuará o arquivamento das mensagens na base de dados, que deverão ser validadas pelos agentes projetados. Nesta tela também será possível solicitar ao sistema que faça a reordenação das regras do sistema, conforme a norma 7 de Lista de Dependências.

A interface 'Cadastro de Regra' possui uma barra de título azul com o ícone de uma pasta e o texto 'Cadastro de Regra'. O corpo da interface é cinza e contém o seguinte layout:

- Um subtítulo 'LISTA DE DEPENDÊNCIA' em azul no topo central.
- Dois campos de entrada: 'Sistema:' com uma lista suspensa selecionando '- nenhum -' e 'Versao:' com um campo de texto vazio.
- Seção 'Regra:' com:
 - Um campo 'Cada' com uma lista suspensa.
 - Dois campos de texto adjacentes, o primeiro vazio e o segundo com uma lista suspensa selecionando '- nenhum -'.
 - Dois campos de texto adjacentes, o primeiro vazio e o segundo com uma lista suspensa selecionando '- nenhum -'.
 - Um campo de texto vazio à esquerda e um botão 'Salvar' à direita.
- Seção 'Regras:' com um grande retângulo branco para a lista de regras.
- Na base da interface, quatro botões alinhados horizontalmente: 'Alterar', 'Analisar', 'Reordenar' e 'Visualizar'.

Figura 112 – Tela de Cadastramento de Regras

5.3.5.4 - Alterar Regra

A tela de alteração de regras possibilitará ao usuário atualizar conceitos pertinentes a cada item envolvido com o sistema que está sendo descrito. Para efeito de alteração somente será permitida a alteração de uma regra por vez, possibilitando assim a verificação das atualizações antes do arquivamento.

Figura 113 – Tela de Atualização de Regras do Sistema

5.3.5.5 - Reordenar Regra

Conforme a regra 7 de Lista de Dependências, todas as frases deverão ser ordenadas pela ordem de importância dos itens informados para o sistema que está sendo projetado. Nesta tela, portanto, o usuário poderá informar uma nova numeração para as frases, garantindo assim a liberdade para que o usuário possa informar a sua ordenação lógica, não aceitando simplesmente a ordenação sugerida por este sistema.

Figura 114 – Tela de reordenação das regras do sistema

5.3.5.6 - Cadastrar Palavra

A tela de cadastro de palavras será utilizada para que sejam cadastradas as palavras de dicionário do sistema que estará sendo utilizado. Neste caso é importante que sejam cadastrados o singular e plural das palavras, sendo que, caso desejado, poderá ainda ser cadastrado o significado da palavra para que se possa ter um dicionário mais completo



A interface de usuário para o cadastro de palavras é exibida em uma janela com o título "Cadastro de Palavras". No topo, o texto "CADASTRO DA PALAVRA" é centralizado. Abaixo, há três campos de entrada: "Palavra:" (um campo de texto único), "Plural:" (um campo de texto único) e "Significado:" (um campo de texto maior). Na base da janela, há uma barra contendo botões para "Novo", "Alterar", "Excluir", "Buscar", "Gravar" e "Cancelar". Abaixo desses botões, há uma segunda linha com botões de navegação: "<<", "<", ">" e ">>", seguidos por um botão "Sair".

Figura 115 – Tela de cadastro de palavras de dicionário do sistema

5.3.5.7 - Palavras Não Atômicas

Conforme é solicitado nas normas de lista de dependências não devem ser utilizadas palavras que indiquem atributos divisíveis, ou melhor, não atômicos. Como esta regra depende uma análise contextual das palavras utilizadas, o sistema somente deverá indicar ao usuário que existe alguma palavra que possivelmente não é atômica. Estas palavras deverão estar cadastradas na base de dados de suporte através desta tela.

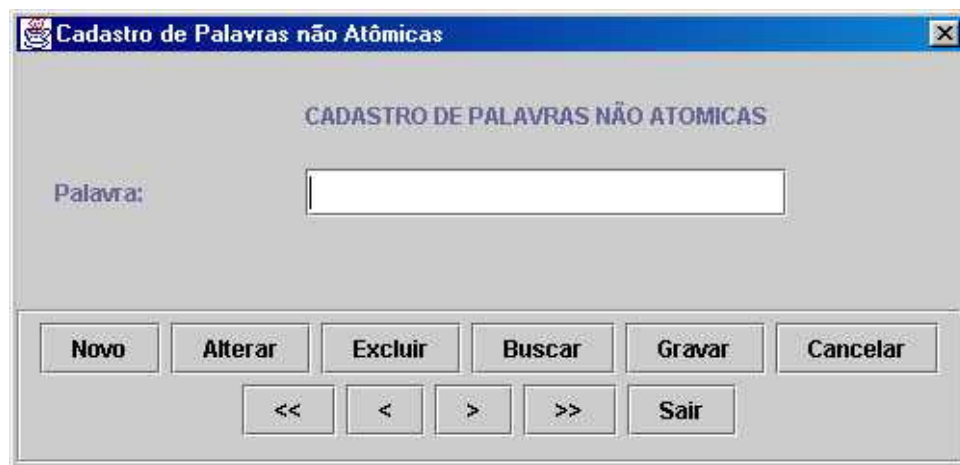


Figura 116 – Tela de cadastro de palavras não atômicas

5.3.5.8 - Cadastrar Sinônimo

Para evitar o uso de palavras que possuam possíveis sinônimos no texto informado pelo usuário, estas palavras deverão estar informadas no sistema para que o sistema possa indicar um possível erro ao usuário.



Figura 117 – Tela de cadastro de sinônimos

5.3.5.9 - Cadastrar Palavra Fixa

A tela de entrada de regras do sistema possui três campos fixos, onde o usuário somente poderá utilizar expressões já cadastradas na base de dados. Ao informar a expressão, o professor deverá indicar se ele é um indicador de facultatividade, ligação com o identificador do sujeito ou apenas verbo de ligação do sujeito com seu predicado.



A janela de software intitulada "Cadastro de Palavras Fixas" apresenta o seguinte layout:

- Barra de título: "Cadastro de Palavras Fixas" com ícone de pasta e botão de fechar.
- Cabeçalho interno: "CADASTRO DE PALAVRAS FIXAS" em letras maiúsculas.
- Campos de entrada:
 - "Palavra:" seguido de um campo de texto curto.
 - "Descrição:" seguido de um campo de texto longo.
 - "Tipo:" seguido de uma lista suspensa com o valor "IF" selecionado.
- Botões de ação na base da janela:
 - Novo
 - Alterar
 - Excluir
 - Buscar
 - Gravar
 - Cancelar
 - Sequência de navegação: <<, <, >, >>
 - Sair

Figura 118 – Tela de palavras das áreas fixas do cadastro de regras do sistema

5.3.5.10 – Mensagens do Sistema

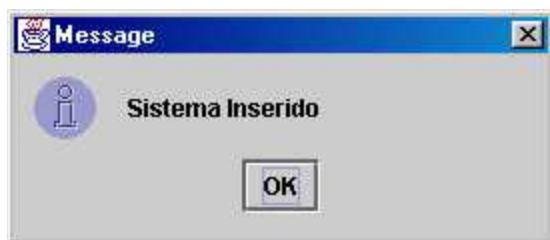


Figura 119 – Mensagem de Criação de um novo sistema.

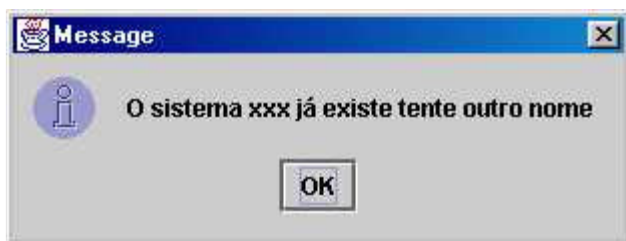


Figura 120 – Mensagem de Erros para sistema com nome duplicado.

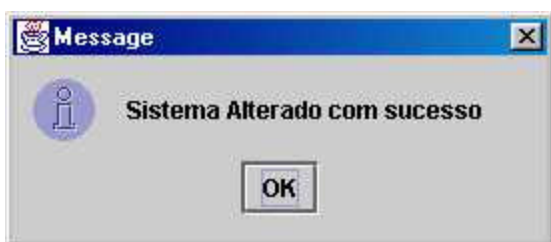


Figura 121 – Mensagem de efetivação de alteração do sistema.

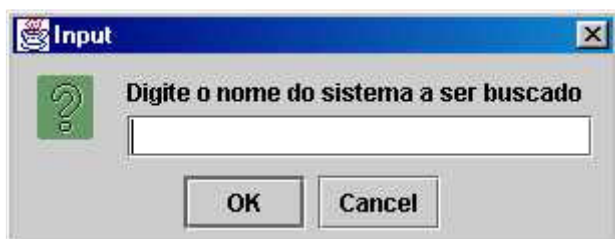


Figura 122 – Tela para entrada de parâmetro de busca de sistema.

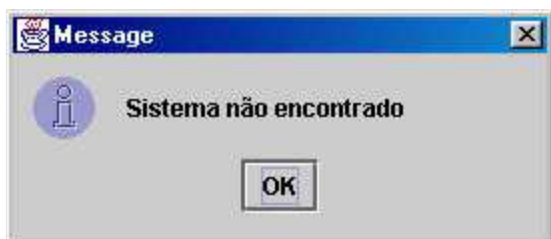


Figura 123 – Tela de erro para parâmetro de busca não encontrado na base de dados



Figura 124 – Tela de erro para campo autor não preenchido no cadastro do sistema.

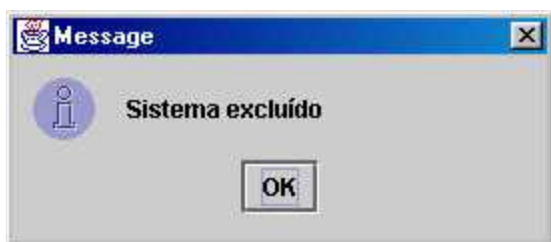


Figura 125 – Tela de informação sobre exclusão do sistema.

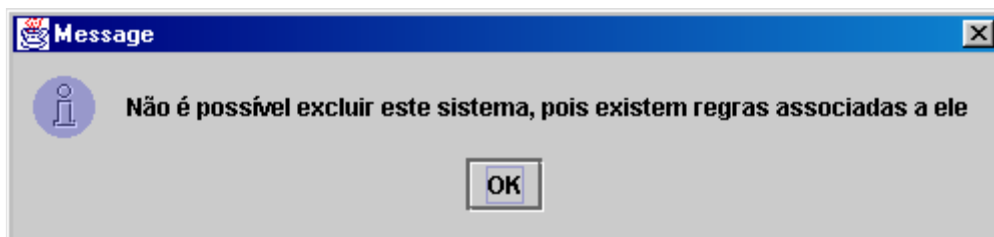


Figura 126 – Tela de informação de erro de integridade referencial entre sistema e regras.

5.4 Documentação da implementação

5.4.1 Descrição da Linguagem de Programação utilizada

A linguagem de programação utilizada é o Java, utilizando-se da máquina virtual JDK 1.3.

Inicialmente o banco de dados utilizado foi Access e atualmente está sendo implementado em Oracle 8i, devido a uma boa interface com o Java.

O sistema será instalado em um único PC ou em vários para uso exclusivo, ou seja mono-usuário.

5.4.2 Descrição da ferramenta utilizada para criação do help on-line

Para a criação do Help On-line foi utilizado o HTML, sendo de grande ajuda na organização das informações referentes ao sistema, com o objetivo de esclarecer as dúvidas referentes ao sistema.

6. – Trabalhos futuros após conclusão.

Como seqüência deste trabalho, algumas novas etapas poderão ser acrescentadas a este projeto, possibilitando assim um melhor uso do software e aproveitando os recursos tecnológicos para aferir toda a técnica de Lista-Diagrama. As etapas seriam as seguintes:

- a) Conversão do Sistema para Internet;
- b) Geração do Diagrama de Dependências;
- c) Correção do Diagrama de dependências com a respectiva retro-alimentação na lista de dependências;
- d) Geração das Tabelas básicas relacionadas (diagrama);
- e) Geração do Diagrama entidade-relacionamento;
- f) Geração dos esquemas para criação das tabelas em uma base de dados, como, por exemplo, Oracle.
- g) Criação de relatórios intermediários do sistema, possibilitando ao professor uma visão da evolução do aluno a cada etapa.
- h) Correção ortográfica da Lista de Dependências.

7. – Conclusão

O uso do software desenvolvido por duas turmas do segundo ano do curso de processamento de dados mostrou-se muito eficiente no que diz respeito à formatação textual das análises de sistemas, fazendo com que os alunos tenham uma técnica de análise mais clara e precisa, evitando redundâncias e facilitando o entendimento dos conceitos do sistema que se deseja desenvolver.

Observou-se também a reflexão sobre os conceitos que estão sendo propostos pelo aluno em sua análise, através dos avisos que o software emite, fazendo com que os alunos tenham mais confiança na sua análise e revisem com mais atenção o que foi escrito, inclusive erros ortográficos, apesar do software não questioná-los.

Outro resultado importante obtido foi observado no momento da criação dos diagramas do sistema, onde o aluno apresentou muito mais eficiência e criatividade nas soluções propostas por conhecer muito bem as regras de negócio do seu sistema.

O software apresenta ainda alguns problemas em tempo de execução, apontados principalmente em sistemas como muitas regras, pois, no momento da verificação contextual existe um grande tempo de espera, principalmente em computadores de menor poder de processamento ou pouca memória RAM. Este problema deverá ser sanado migrando-se este software para uso através da Internet, com uma arquitetura de Servlets e Banco de Dados Oracle, onde todo o poder de processamento necessário será ofertado pelo servidor da aplicação.

Bibliografia

+IBM – Via Voice – Manual do Sistema

+UNISYS – Material de Bussiness Desing - 1989

CHEN, Peter. *Modelagem de Dados*. São Paulo: Makron Books, 1990, 80p.

BAÑOS, René Martinez. *Apostila sobre Lista Diagrama*. Londrina: 1997, Apostila, 80p.

DATE, C.J. *Introdução a Sistemas de BANCO DE DADOS – Traduzido da 7a Edição Americana*. São Paulo: Editora Campus, 2001, 670p.

ELMASRI, Ramez, NAVATHE e Shamkant B. *Fundamentals of database systems*. 3.ed. Addison-Wesley, 2000, 955p.

FELICIANO NETO, Acácio, FURLAN, José David e HIGA, Wilson. *Engenharia da Informação: Metodologia, Técnicas e Ferramentas*. São Paulo: McGraw-Hill, 1988, 262p.

FURLAN, Davi José. *Modelagem de Objetos através da UML – The Unified Modeling Language*. São Paulo: Makron Books, 1998, 329p.

HEUSER, Carlos A. *Projeto de Banco de Dados*. Porto Alegre: Ed. Sagra Luzzatto, 1999, 216p.

LEVINE, Robert I., DRANG, Diane E. e EDELSON, Barry. *Inteligência artificial e sistemas especialistas: Aplicações e exemplos práticos*. São Paulo: McGraw-Hill, 1988, 264p.

MACHADO, Felipe Nery R., ABREU, Maurício. *Projeto de Banco de Dados, uma visão prática*. São Paulo: Editora Érica, 1995, 298p.

MARTIN, James. *Engenharia da Informação – Introdução*. São Paulo: Editora Campus, 1991, 196p.

RICH, Elaine, KNIGHT, Kevin. *Artificial Intelligence*. Estados Unidos: McGraw-Hill, 1991, 621p. 2ed.

SHOVAL, P. and SHREIBER, N., *Database Reverse Engineering: From the Relational to the Binary Relationship Model, Data & Knowledge Engineering*. Elsevier Science 293-315. Outubro/1993

SILBERSCHATZ, Abraham, KORTH, Henry F. e SUDARSHAN, S. *Sistema de Banco de Dados*. 3.ed. São Paulo: Makron Books, 1999, 777p.

TORRE, Marina Della. *Inteligência Artificial*. [online] Disponível na Internet via

WWW. URL: <http://fly.to/bytegirl>. Arquivo capturado em 01 de Abril de 2001.

ULLMAN, Jeffrey D. *Principles of Database and Knowledge-Base Systems*. New York: Computer Science Press, 1988, 460p.

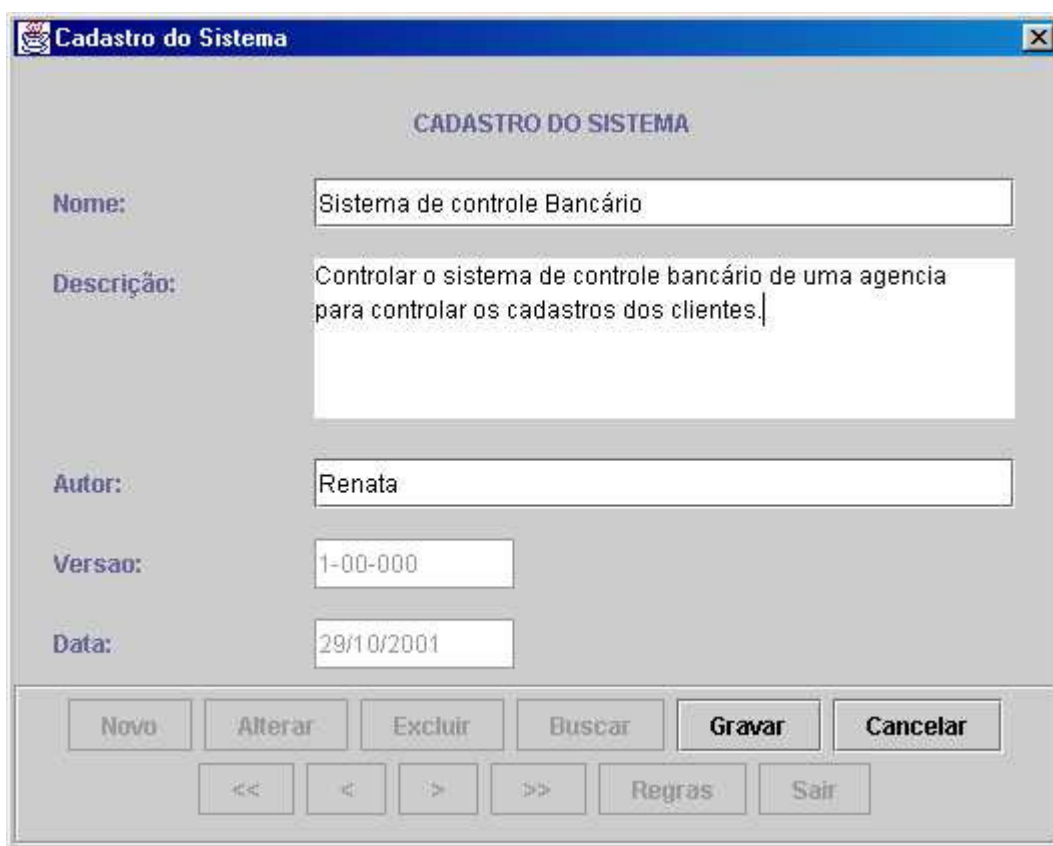
ULLMAN, Jeffrey D. e WIDOM, Jennifer. *A First Course in Database Systems*. New Jersey: Prentice-Hall, 1997, 470p.

VALLIN, Celso. *Proibido Dar Aula*. [online]: Disponível na Internet via WWW. URL <http://www.edutecnet.com.br/> Capturado em Fev/2001.

ANEXOS

Anexo A – Impressão do lay-out das telas e relatórios

Cadastrar Sistema



The screenshot shows a Windows-style window titled "Cadastro do Sistema". Inside, the title "CADASTRO DO SISTEMA" is centered. Below it, there are five labeled input fields: "Nome:" with the value "Sistema de controle Bancário", "Descrição:" with the text "Controlar o sistema de controle bancário de uma agencia para controlar os cadastros dos clientes.", "Autor:" with the value "Renata", "Versao:" with the value "1-00-000", and "Data:" with the value "29/10/2001". At the bottom, there are two rows of buttons. The first row contains "Novo", "Alterar", "Excluir", "Buscar", "Gravar", and "Cancelar". The second row contains navigation buttons "<<", "<", ">", ">>", "Regras", and "Sair".

Nome:	Descrição:	Autor:	Versao:	Data:
Sistema de controle Bancário	Controlar o sistema de controle bancário de uma agencia para controlar os cadastros dos clientes.	Renata	1-00-000	29/10/2001

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Regras Sair

Cadastrar Regras

Cadastro de Regras Regra

LISTA DE DEPENDÊNCIA

Sistema: Sistema controle bancário Versao: 1-00-000

Regra: Cada Banco é identificado

pelo seu número e tem como atributos

nome, rua, endereço

Salvar

Regras:

1 - Cada Cliente é identificado pelo CPF e tem como atributos nome, telefone, rua, número.

Alterar Analisar Reordenar Visualizar

Cadastrar Palavras

Cadastro de Palavras

CADASTRO DA PALAVRA

Palavra: a Plural: as

Significado: Artigo feminino

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

Cadastrar Palavras Fixas

Cadastro de Palavras Fixas

CADASTRO DE PALAVRAS FIXAS

Palavra:

Descrição:

Tipo:

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

Cadastrar Palavras Não Atômicas

Cadastro de Palavras não Atômicas

CADASTRO DE PALAVRAS NÃO ATOMICAS

Palavra:

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

Cadastrar Sinônimos

Cadastro de Sinônimos

CADASTRO DE SINONIMOS

Palavra: pagar

Sinônimo: quitar

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

Visualização do Relatório de Regras

Visualizar Regra

LISTA DE DEPENDÊNCIA

SISTEMA: Sistema controle bancário
DESCRIÇÃO:
AUTOR: Renata
VERSÃO: 1-00-000 DATA: 29/10/2001

1 - Cada Cliente é identificado pelo CPF e tem como atributos nome, telefone, rua, número.
2 - Cada banco é identificado pelo seu número e tem como atributos nome.

Imprimir Imprimir Original

Impressão do Relatório de Regras

LISTA DEPENDÊNCIA

SISTEMA: Sistema controle bancário

DESCRIÇÃO:

AUTOR: Renata

VERSÃO: 1-00-000

DATA: 29/10/2001

1 - Cada Cliente é identificado pelo CPF e tem como atributos nome, telefone, rua, número.

2 - Cada banco é identificado pelo seu número e tem como atributos nome.

Impressão do Relatório de Regras Original

RELATÓRIO ORIGINAL – LISTA DEPENDÊNCIA

SISTEMA: Sistema controle bancário

DESCRIÇÃO:

AUTOR: Renata

VERSÃO: 1-00-000

DATA: 29/10/2001

1 - Cada Cliente é identificado pelo CPF e tem como atributos nome, telefone, rua, número.

2 - Cada banco é identificado pelo seu número e tem como atributos nome.

Relatório das Palavras não Atômicas

Relatório das Palavras Não Atômicas

PALAVRA NÃO ATÔMICA

endereço

endereço

Relatório das Palavras

Relatório das Palavras		
PALAVRA	SIGNIFICADO	PLURAL
a	Artigo indicativo de Feminino	as
o	Artigo Definido Masculino	os

Relatório dos Sinônimos

Relatório de Sinonimos	
PALAVRA	SINONIMO
pagar	quitar

Relatório dos Sistemas

Relatório dos Sistemas

SISTEMA

AUTOR

Sistema controle bancário

Renata

Anexo B – Manual de Instalação

A vertical, light blue, semi-transparent graphic of a circuit board with various traces and components, positioned on the left side of the page.

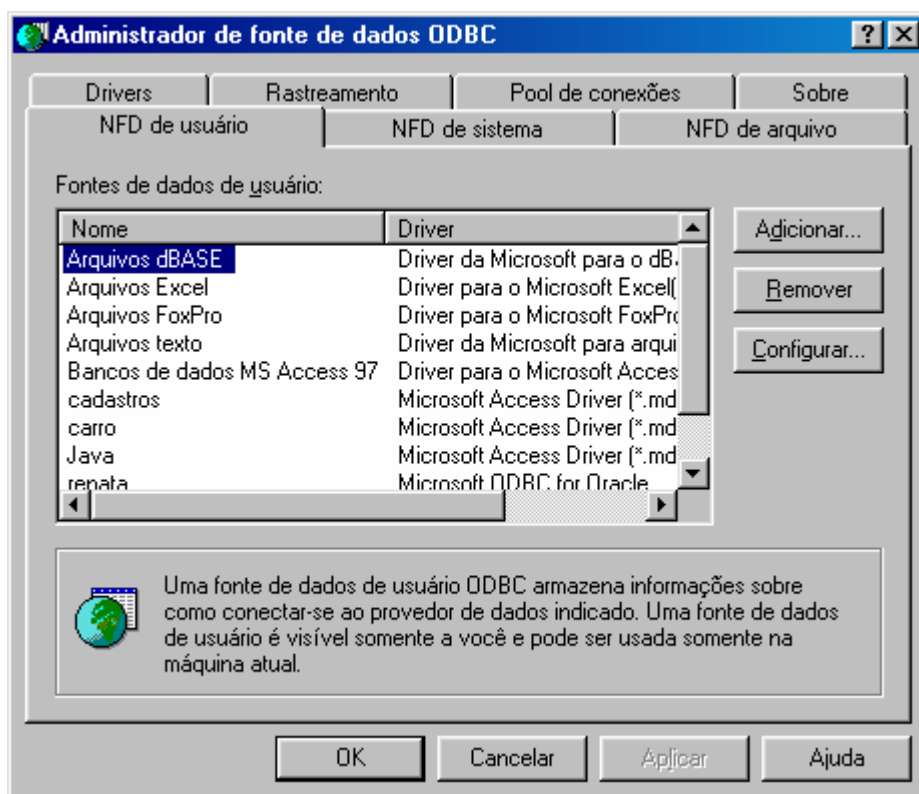
MANUAL DE INSTALAÇÃO

Software de Análise Conceitual

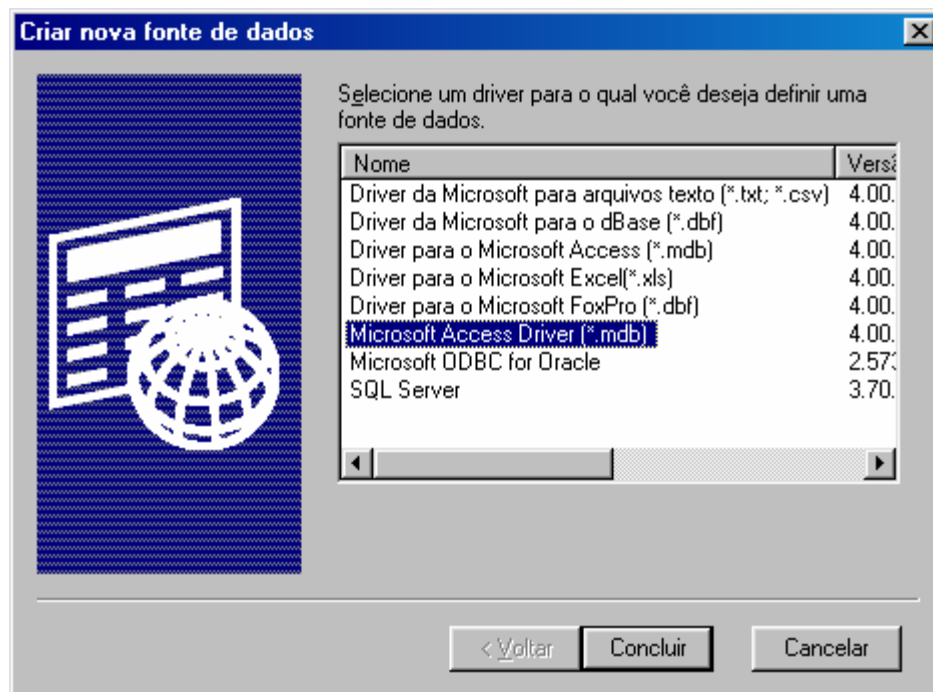
UTILIZANDO BANCO DE DADOS ACCESS

Requisitos para Instalação: Para a instalação desta versão será necessário o micro estar com a instalação da máquina virtual Java (JDK1.3 ou superior) e ter instalado também o Access 97 ou superior.

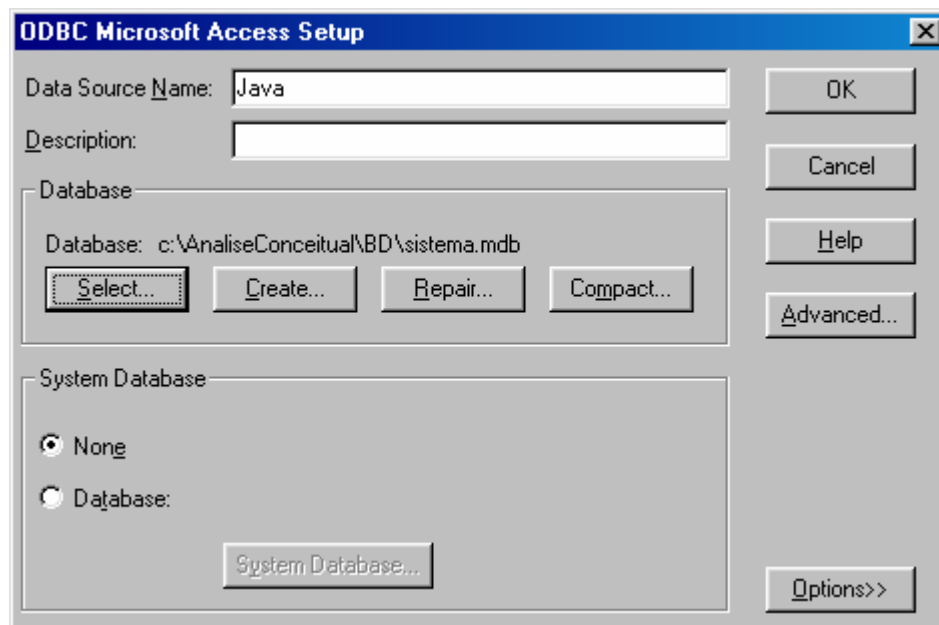
1. Descompactar o arquivo InstAcs.zip em c:\AnaliseConceitual
2. Clicar em Iniciar / Configurações / Painel de controle / Fontes de dados ODBC



3. Clicar em adicionar
4. Escolher Microsoft Access Driver (*.mdb) / Concluir



5. No campo Data Source Name preencha com Java e no Database clique em select e selecione o caminho c:\AnaliseConceitual\BD\sistema.mdb

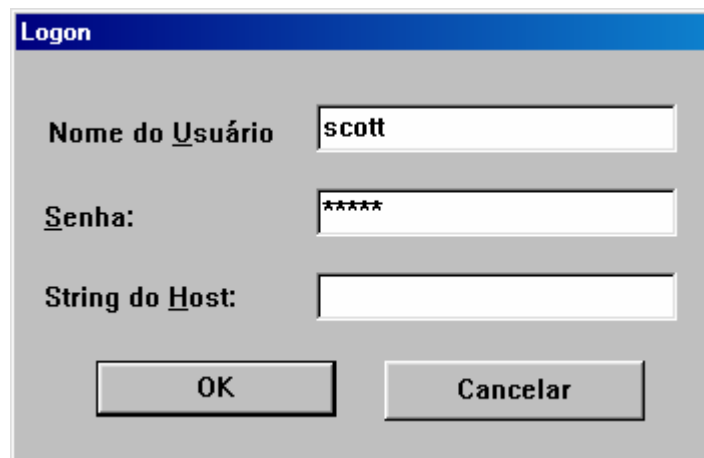


6. Clique com o botão direito na área de trabalho, escolha Novo / Atalho
7. Na linha de comando digite: "C:\AnaliseConceitual\Lista Diagrama.exe"
8. Clique em avançar.

UTILIZANDO BANCO DE DADOS ORACLE

Requisitos para Instalação: Para a instalação desta versão será necessário o micro estar com a instalação da máquina virtual Java (JDK1.3 ou superior) e ter instalado também o banco de dados Oracle 8i na raiz, ou seja em c:\Oracle.

1. Descompactar o arquivo InstOra.zip em c:\AnaliseConceitual.
2. Executar o SQL Plus e se logar com scott e senha tiger.



The image shows a Windows-style dialog box titled "Logon". It has a blue title bar. Inside, there are three labeled text input fields. The first is labeled "Nome do Usuário" and contains the text "scott". The second is labeled "Senha:" and contains six asterisks "*****". The third is labeled "String do Host:" and is currently empty. At the bottom of the dialog, there are two buttons: "OK" on the left and "Cancelar" on the right.

3. Digitar a seguinte linha de comando @c:\AnaliseConceitual\table.
4. Clique com o botão direito na área de trabalho, escolha Novo / Atalho.
5. Na linha de comando digite: "C:\AnaliseConceitual\Lista Diagrama.exe".
6. Clique em avançar.

Após a instalação verificar o manual do usuário o funcionamento do sistema.

Obs.: Todo manual foi feito baseado no sistema operacional Windows 98, podendo sofrer algum tipo de alteração em outros sistemas operacionais.

Anexo C – Manual do usuário



MANUAL DO USUÁRIO

Este manual exemplifica detalhadamente como operar o software de análise conceitual, exemplificando de maneira clara e objetiva todas as suas funções do mesmo.

Conhecendo o sistema

O software de análise conceitual permite com que o usuário entre com os dados do levantamento do sistema para que com isso o sistema possa analisar e identificar possíveis falhas ou dúvidas.



Lista Diagrama

Se a instalação do software já foi concluída, clique no ícone Lista Diagrama em sua área de trabalho.

Fazendo isso você já poderá iniciar seu sistema.

Utilizando o sistema

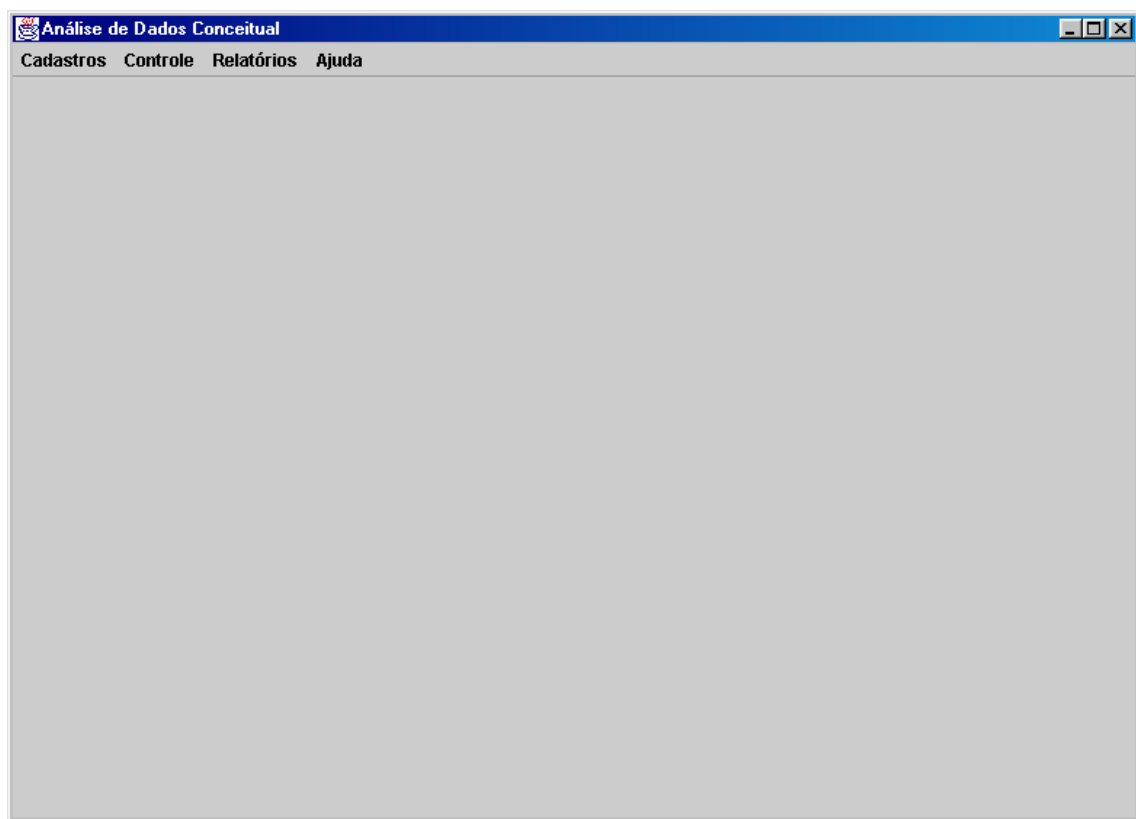
O manuseio do sistema é muito simples, inicialmente todas as telas com exceção a do cadastro de regras são padronizadas.

Vejamos a seguir a função dos botões que ficam no canto superior das telas:

Este navegador é padronizado, portanto será o mesmo em quase todas as telas.

Menu Principal

O menu principal traz de forma organizada todas telas do sistema, separadas por tópicos: cadastro, controle, relatórios e ajuda.



No menu de Cadastros teremos os cadastros de sistema, de regras e a disponibilidade de deixar o sistema com Sair.

No menu de controle trataremos os controles de palavras, palavras não atômicas, palavras fixas e sinônimos.

No menu relatórios será disponibilizado os relatórios dos sistemas existentes, das palavras não atômicas cadastradas, do relatório das palavras e do relatório dos sinônimos

No menu ajuda teremos o conteúdo de ajuda descrito aqui e no item sobre teremos informações sobre o sistema desenvolvido.

Cadastrar um sistema

O cadastro do sistema, faz com que você crie uma instância para inicializar o processo de análise conceitual.

Para cadastrar um sistema é necessário ir ao menu cadastrar e escolher a opção cadastrar sistema, abrirá a seguinte tela:

Cadastro do Sistema

CADASTRO DO SISTEMA

Nome:

Descrição:

Autor:

Versão:

Data:

Novo **Alterar** **Excluir** **Buscar** **Gravar** **Cancelar**

<< **<** **>** **>>** **Regras** **Sair**

No campo nome, deve ser digitado o título de sua análise.

No campo descrição, deve ser descrito uma breve prévia do análise que será feito.

No campo autor, deve ser digitado o nome do desenvolvedor do sistema.

Os campos versão e data estarão desabilitados, pois o sistema irá preencher e atualizar automaticamente.

Cadastrar uma Regra

No cadastro de regras será analisada a validação de suas frases e se realmente estará sendo aplicado as técnicas de lista diagrama, cumprindo suas regras.

Antes de cadastrar uma regra, deve-se haver um sistema cadastrado para fazer referência a mesma. Há duas maneiras de se cadastrar regras. Vejamos a seguir:

Uma das opções é no momento do cadastro do sistema, onde existe o botão Regras, o qual trará a tela de cadastros de regra do sistema selecionado.



A imagem mostra uma janela de software intitulada "Cadastro do Sistema". No topo, há uma barra azul com o título e ícones de minimizar, maximizar e fechar. O conteúdo principal da janela tem um fundo cinza claro e o título "CADASTRO DO SISTEMA" no topo central. Abaixo do título, há cinco campos de entrada rotulados: "Nome:" (um campo de texto simples), "Descrição:" (um campo de texto maior), "Autor:" (um campo de texto simples), "Versão:" (um campo de texto simples) e "Data:" (um campo de texto simples). Na base da janela, há uma barra contendo botões de ação: "Novo", "Alterar", "Excluir", "Buscar", "Gravar" e "Cancelar" na primeira linha; e "<<", "<", ">", ">>", "Regras" e "Sair" na segunda linha.

Uma outra maneira seria no próprio menu principal, no item Cadastros / Cadastro de Regras.

Ambas maneiras trarão a tela abaixo relacionada:

Cadastro de Regra

LISTA DE DEPENDÊNCIA

Sistema: - nenhum - Versao:

Regra: Cada - nenhum -

Salvar

Regras:

Alterar Analisar Reordenar Visualizar

Ao entrar com a frase e salvá-la o sistema fará uma verificação na mesma e em cada frase anterior, auxiliando então caso exista alguma inconsistência na frase, sugerindo modificações e permitindo a progressão ou não.

Para a alteração de uma regra é muito simples, ao clicar em alterar, será mostrado a seguinte tela:

Alterar Regra

ALTERAÇÃO NA LISTA DE DEPENDÊNCIA

Sistema: - nenhum -

Regra:

Cada - nenhum -

- nenhum -

Anterior **Próximo** **Confirmar** **Excluir**

Nesta tela poderá se feita qualquer alteração no sistema no que se diz respeito aos dados. O sistema trará automaticamente os dados do primeiro registro baseado na escolha do sistema na tela de regras.

Para reordenar uma regra basta clicar em Reordenar e lhe será apresentada a seguinte tela:

Reordenar Regras

REORDENA LISTA DE DEPENDÊNCIA

Sistema: - nenhum -

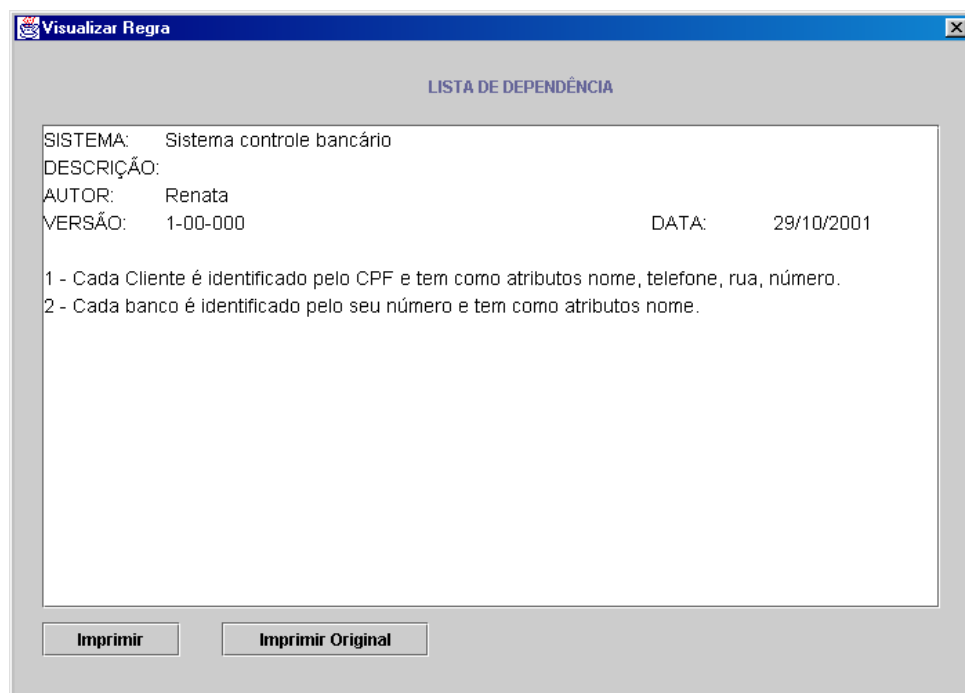
Regra:

☐

Anterior Próximo - + Confirmar

Nesta tela poderá ser utilizado os botões de – ou + para diminuir ou aumentar a posição da frase no sistema. Toda vez que é feito uma

alteração é necessário confirmar, senão a alteração não será possível.



Para visualizar o relatório final da regra basta clicar no botão visualizar que fará com que seja mostrado a descrição do sistema e das regras referentes.

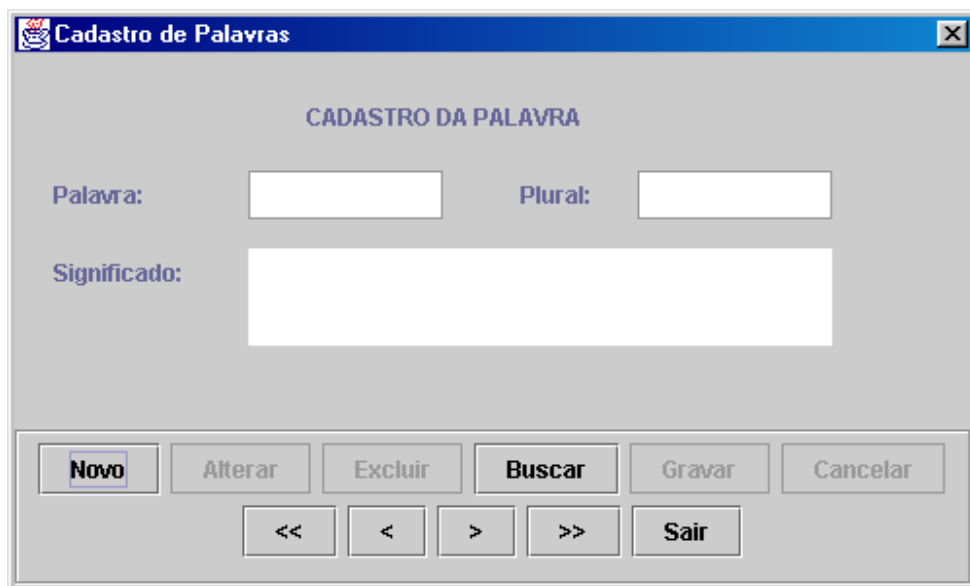
O botão imprimir permite a impressão do relatório conforme mostrado na tela.

O botão imprimir original permite a impressão do relatório das regras originais desde a elaboração do sistema a exclusão de frases

Cadastrar Palavra

O cadastro de uma palavra é um cadastro básico servindo de referência para as palavras cadastradas no sistema.

Para cadastrar uma palavra é necessário ir ao menu controle e escolher a opção palavra, aparecendo a seguinte tela:



The screenshot shows a software window titled "Cadastro de Palavras". Inside the window, the title "CADASTRO DA PALAVRA" is centered at the top. Below the title, there are three input fields: "Palavra:" followed by a single-line text box, "Plural:" followed by a single-line text box, and "Significado:" followed by a multi-line text box. At the bottom of the window, there is a row of six buttons: "Novo", "Alterar", "Excluir", "Buscar", "Gravar", and "Cancelar". Below this row, there is another row of five buttons: "<<", "<", ">", ">>", and "Sair".

O campo palavra deve ser preenchido de acordo com a palavra que deverá ser analisada.

No campo plural deve ser preenchido referente a palavra cadastrada.

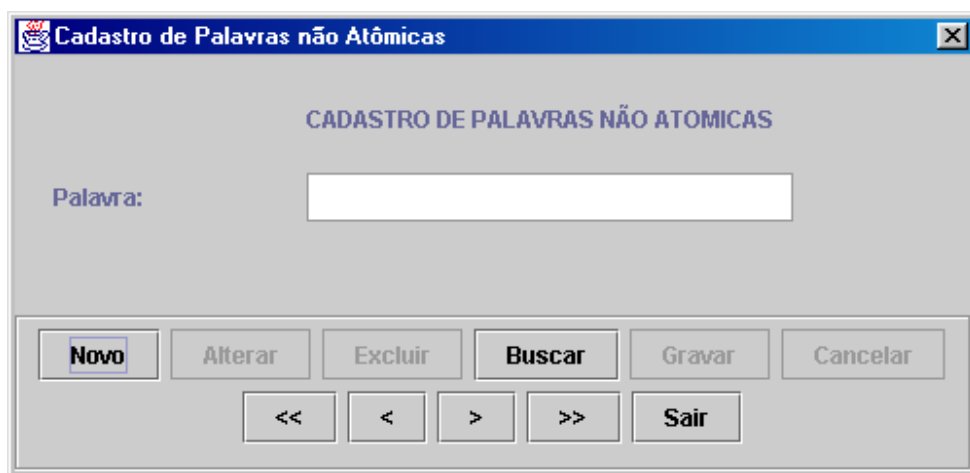
No campo significado deve ser preenchido conforme o significado da palavra cadastrada.

Cadastrar uma Palavra não Atômica

Para o cadastro de uma palavra não atômica o processo também é o mesmo que o cadastro de palavra, visando que uma palavra não atômica não pode estar em

ocorrência com a frase da lista diagrama. Uma palavra não atômica não possui sentido único.

No menu principal, clicando em controlar palavra não atômica aparecerá a seguinte tela:

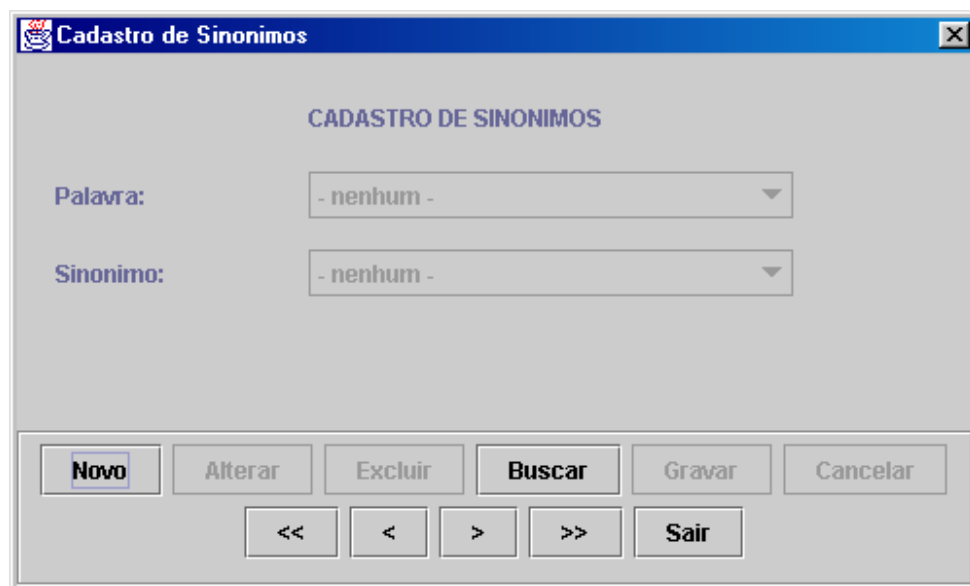


No campo palavra deve ser preenchido com palavras que não fazem um sentido completo e único.

Cadastrar um sinônimo

O cadastro de um sinônimo também é um cadastro básico servindo de como base nas validações das regras de negócio.

Para cadastrar um sinônimo é necessário ir ao menu controle e escolher a opção sinônimo, logo em seguida aparecerá a seguinte tela:



Cadastro de Sinonimos

CADASTRO DE SINONIMOS

Palavra: - nenhum -

Sinonimo: - nenhum -

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

No campo palavra deve ser relacionado a palavra em que deseja se cadastrar o sinônimo.

No campo sinônimo é preenchido relacionado ao campo palavra.

Cadastrar uma palavra Fixa

Cadastro de Palavras Fixas

CADASTRO DE PALAVRAS FIXAS

Palavra:

Descrição:

Tipo: IF ▼

Novo Alterar Excluir Buscar Gravar Cancelar

<< < > >> Sair

Para o cadastro de uma palavra fixa o procedimento é o mesmo que os cadastros anteriores, lembrando-se apenas que o tipo de palavra fixa se refere a determinados pontos da lista diagrama.

No campo Palavra deve ser preenchido com a palavra fixa que deverá aparecer no cadastro de regras.

No campo descrição deve ser descrito o significado da palavra cadastrada.

No campo tipo deve ser preenchido por IF, LI e VR que significam respectivamente Identificador Facultativo, Ligação e verbo, onde estarão relacionadas a cada frase e em cada ponto específico.

Regra: Cada ▼

- nenhum - ▼

- nenhum - ▼

Salvar

